

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STROJNÍCKA FAKULTA**

**EXPERIMENTÁLNE ZARIADENIE NA
RIADENIE STATICKEJ DEFORMÁCIE
VOTKNUTÉHO NOSNÍKA**

Bakalárska práca

SjF-13432-87733

2019

Eva Vargová

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STROJNÍCKA FAKULTA**

**EXPERIMENTÁLNE ZARIADENIE NA
RIADENIE STATICKEJ DEFORMÁCIE
VOTKNUTÉHO NOSNÍKA**

Bakalárska práca

SjF-13432-87733

Študijný odbor: Automatizácia a informatizácia strojov a procesov
Študijný program: 5.2.14 automatizácia
Školiace pracovisko: Ústav automatizácie, merania a aplikovanej informatiky
Vedúci záverečnej práce: doc. Ing. Gergely Takács, PhD.
Konzultant: doc. Ing. Martin Gulan, PhD.

Bratislava, 2019

Eva Vargová



ZADANIE BAKALÁRSKEJ PRÁCE

Študentka: **Eva Vargová**
ID študenta: 87733
Študijný program: automatizácia a informatizácia strojov a procesov
Študijný odbor: 5.2.14. automatizácia
Vedúci práce: doc. Ing. Gergely Takács, PhD.
Konzultant: doc. Ing. Martin Gulan, PhD.
Miesto vypracovania: ÚAMAI SjF, STU v Bratislave

Názov práce: **Experimentálne zariadenie na riadenie statickej deformácie votknutého nosníka**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Úlohou študenta je navrhnuť, realizovať a vyrobiť rozširovací modul pre prototypizačnú platformu Arduino v rámci open-source projektu „AutomationShiled“. Jedná sa o návrh miniaturizovaného laboratórneho experimentu riadenia statickej deformácie votknutého nosníka cez servomotor, kde údaje o deformácii nosníka budú zbierané cez odporový tenzometer. Práca musí obsahovať ovládaci softvér a inštrukčné príklady. Študent navrhne plošný spoj v CAD prostredí DipTrace, vytvorí programátorské rozhranie (API) v jazyku C/C++ pre Arduino IDE, manažuje verzie projektu v Git pre GitHub a píše dokumentáciu v jazyku Markdown.

Rozsah práce: cca. 30-50 s.

Riešenie zadania práce od: 11. 02. 2019
Dátum odovzdania práce: 24. 05. 2019

Eva Vargová
študentka

prof. Ing. Cyril Belavý, CSc.
vedúci pracoviska

Slovenská technická univerzita
v Bratislave
Dekanát Strojníckej fakulty
Útvor pedagogických činností
812 31 Bratislava, Nám. Slobody 17

prof. Ing. Cyril Belavý, CSc.
garant študijného programu

Čestné prehlásenie

Vyhlasujem, že som záverečnú prácu vypracovala samostatne s použitím uvedenej literatúry.

Bratislava, 24. mája 2019

.....
Eva Vargová

Ďakujem vedúcemu bakalárskej práce, doc. Ing. Gergely Takács, PhD., za veľkú trpežlivosť a odbornú pomoc pri vyhotovení bakalárskej práce. Chcela by podčakovať aj konzultantom bakalárskej práce, doc. Ing. Martinovi Gulánovi, PhD., za pomoc s hardvérovou časťou a Ing. Erikovi Mikulášovi za ochotu a rady v oblasti 3D tlače. A na záver by som sa chcela podčakovať mojej rodine a priateľovi za neuveriteľnú podporu po celý čas písania bakalárskej práce.

Bratislava, 24. mája 2019

Eva Vargová

Názov práce: Experimentálne zariadenie na riadenie statickej deformácie votknutého nosníka

Kľúčové slová: Arduino, TugShield, riadenie, deformácia, servo motor

Abstrakt: Úvodom bakalárskej je popísaná situácia vo svete ohľadom implementácie Arduina do učebných osnov. Arduino využívajú viaceré projekty, zaobrájúce sa vzdelávaním v oblasti teórie riadenia. Druhá kapitola sa venuje vytvoreniu vhodného hardvéru pre experimentálne zariadenie. Je tu zahrnutý aj opis a parametre použitých komponentov, vývoj prototypov a návrh finálneho technického diela. Softvérové metódy na ovládanie hardvéru sú opísané v tretej kapitole. Trieda TugShield obsahuje štyri metódy na inicializovanie a kalibráciu hardvéru, čítanie výstupných hodnôt zo snímača ohybnosti a zapisovanie vstupných hodnôt do servo motora. V poslednej kapitole sú uvedené demonstratívne príklady, ktoré testujú hardvér a vytvorené programové rozhranie. Výsledkom práce je vytvorenie didaktickej pomôcky pre cvičenia z teórie riadenia pomocou lacného hardvéru.

Title: Experimental device for controlling the static deformation of a cantilever beam

Keywords: Arduino, TugShield, control, deformation, servo motor

Abstract: The introduction of the bachelor's thesis describes the state-of-the-art regarding the implementation of Arduino into the engineering curricula. Arduino is used in variety ways in the didactic process of control engineering. The second chapter deals with the constructing the hardware of the experimental equipment. The description and parameters of the components are included in this chapter as well, along with the development of the prototypes and the design of the final product. Software methods for controlling the hardware are described in Chap. 3. The TugShield class, which was created within this thesis, contains four methods for initializing and calibrating hardware, reading output values from the flexibility sensor, which is the main part of the TugShield and writing input values to the servo motor. There are demonstrative examples that test hardware and the created API in the last chapter. The result of this thesis is a hovel didactic tool for control systems engineering education using low-cost hardware.

Obsah

Úvod	1
1 Arduino ako didaktická pomôcka	2
1.1 Súčasná situácia	2
1.2 Projekty vo svete	3
1.3 Projekt AutomationShield	6
2 Tvorba hardvéru	7
2.1 Popis a parametre použitých komponentov	7
2.1.1 Prototypová doska pre Arduino UNO	7
2.1.2 Komponenty vyrobené pomocou 3D tlače	7
2.1.3 Operačný zosilňovač	8
2.1.4 Rezistory	9
2.1.5 Potenciometer	10
2.1.6 Nosník	10
2.1.7 Servo motor	11
2.1.8 Snímač ohybnosti - flexi senzor	12
2.1.9 Kondenzátor	13
2.1.10 Dióda	13
2.1.11 Zenerova dióda	13
2.2 Vývoj prototypov	14
2.2.1 Prvý návrh	14
2.2.2 Druhý návrh	15
2.2.3 Tretí návrh	17
2.3 Vytvorenie plošnej dosky	19
2.4 Konečný model TugShieldu	24
3 Tvorba softvéru	26
3.1 Trieda TugShield	26
3.2 Metóda Begin	27
3.3 Metóda Calibration	28
3.4 Metóda ActuatorWrite	29
3.5 Metóda SensorRead	30

4 Demonstračné príklady	31
4.1 Príklad identifikácie systému TugShield	31
4.2 Príklad s aplikovaním PID regulátora.	32
4.3 Príklad Selftest	34
5 Záver	38
5.1 Návrhy na zlepšenie	38
Literatúra	40

Zoznam obrázkov

1.1	Celkový potrebný hardvér na prototyp meniča [15].	4
1.2	Simulácia energetickej spotreby štyroch mestských častí pomocou Arduina [23].	4
1.3	Výhody Arduina v edukačnom systéme [22].	5
2.1	Komponenty vytlačené pomocou 3D tlače.	8
2.2	Schematická značka operačného zosilňovača.	8
2.3	Vnútorné rozhranie operačného zosilňovača.	9
2.4	Schematická značka rezistora.	9
2.5	Schematická značka potenciometra.	10
2.6	Rozmery nosníka.	11
2.7	Schematická značka servo motora.	12
2.8	Schematická značka flexi snímača.	12
2.9	Schematická značka kondenzátora.	13
2.10	Schematická značka diódy.	14
2.11	Schematická značka zenerovej diódy.	14
2.12	Prvý prototyp.	15
2.13	Zapojenie neinvertujúceho operačného zosilňovača.	15
2.14	Druhý prototyp.	16
2.15	Tretí prototyp.	17
2.16	Signál zo snímača bez zosilnenia.	19
2.17	Slabé zosilnenie signálu.	19
2.18	Saturovanie signálu.	19
2.19	Vhodné zosilnenie.	19
2.20	Najvhovujúce zosilnenie.	20
2.21	Schematický symbol operačného zosilňovača.	21
2.22	Schéma zapojenia.	22
2.23	Rozmiestnenie komponentov na PCB dosku.	23
2.24	Predná a zadná strana finálnej PCB dosky.	23
2.25	PCB doska TugShieldu.	24
2.26	Finálny produkt.	25
4.1	Výstup z príkladu identifikácie systému.	33
4.2	Výstup z príkladu PID.	35
4.3	Výstup z príkladu PID pri pozícii serva väčších ako 35°	36
4.4	Výstup z príkladu Self-Test.	37

Zoznam tabuliek

2.1	Chemické zloženie ocele AISI 301[19]	11
2.2	Fyzikálne vlastnosti ocele AISI 301[19]	11
2.3	Cena a informácie o použitých komponentov	24

Úvod

Súčasný problém skoro všetkých vzdelávacích inštitúcií je nedostatok finančných zdrojov a personálu na prevádzkovanie laboratórnych experimentov. Študenti sú nútení vykonávať pokusy len v laboratórnych priestoroch pod prísnym dohľadom vyučujúcich a s maximálnou opatrnosťou.

Táto situácia nie je príliš priaznivá pre študentov so zameraním na automatizáciu. Na čerstvých absolventov sa kladú vysoké požiadavky a očakáva sa od nich, že budú vedieť realizovať návrhy riadenia späťnej väzby rovno na hardvéri.

Príchodom Arduina na trh vzniká perspektívna východisková situácia pri riešení spomínaných problémov. Arduino, ktorého základom je mikroradičová jednotka, môže byť použité pri realizácii spätno-väzobného riadenia alebo pri vytváraní algoritmov pre automatické riadenie.

Za účelom implementovať Arduino do vzdelávacieho systému sa vytvoril projekt s názvom AutomationShield. Vytvára prídavný hardvér, takzvané štíty, ktoré rozširujú možnosti využitia základnej mikroradičovej prototypizačnej dosky. Projekt vytvára aj softvér na riadenie štítov.

Úvodom bakalárskej práce je spomenutá situácia vo vzdelávacích inštitúciách vo svete. Prezentované sú projekty z rôznych krajín, v ktorých hlavný komponent je Arduino. Sú to originálne, ale ľažko reprodukované projekty, kvôli nedostačujúcej dokumentácií. Spomenutá je aj situácia na Slovensku.

Druhá kapitola sa venuje hardvérovej časti experimentálneho zariadenia na riadenie statickej deformácie votknutého nosníka s pracovným názvom TugShield. Charakterizuje funkcie, opis a parametre použitých komponentov, vývoj prototypov a návrh finálneho TugShieldu. Kapitola rieši aj návrh zosilnenia operačného zosilňovača, návrh schémy zapojenia a plošnej dosky.

Softvérové metódy na ovládanie hardvéru TugShieldu sú uvedené v tretej kapitole. Knižnica pre štít obsahuje triedu s názvom TugShield, ktorá obsahuje štyri metódy: inicializovanie hardvéru, kalibrácia, prečítanie hodnoty zo snímača ohybnosti a zapísanie polohy ramena servo motora.

V poslednej kapitole sú popísané demonštračné príklady, ktoré slúžia na otestovanie hardvéru a takisto metód z knižnice. Príklad na identifikáciu systému slúži na získanie vstupno-výstupných dát pomocou vytvorenia pulzného radu. Druhý demonštračný príklad má za úlohu riadiť spätnú väzbu ohybu nosníka pomocou PID regulátora a príklad "Self-test" testuje správnosť kalibrácie a funkčnosť štítu.

1 Arduino ako didaktická pomôcka

1.1 Súčasná situácia

Vysokoškolské vzdelávanie sa veľmi zmenilo od zavedenia vyspelých softvérových nástrojov. Klasické riešenia problémov pomocou papiera a pera dnes už nie sú v móde a boli pretransformované na riešenia pomocou počítačov, kde sa dá daný problém navrhnúť a odsimulovať jeho spätnú väzbu. Mnohé nové technológie sú interaktívne, preto je veľmi dôležité uľahčovať vytváranie prostredí, v ktorých môže byť učenie nie len teoretické, ale aj reálne vykonávanie a prijímanie spätej väzby, ktorými sa zlepší porozumenie a budovanie nových poznatkov [29].

Trh práce kladie vysoké požiadavky na absolventov vysokých škôl so zameraním na automatizáciu. Požiadavky sa nevzťahujú len na študentov, ale samozrejme aj na vyučujúcich a univerzity, v ktorých budúci inžinieri študujú. Od čerstvých absolventov, ktorí nastupujú do zamestnania sa čoraz častejšie očakáva, že budú vedieť realizovať návrhy riadenia spätej väzby rovno na hardvéri [29].

Veľký prelom prišiel príchodom Arduina na trh. Jeho základom je mikroradičová jednotka (MCU), ktorá je vhodná na algoritmy a realizáciu spätno-väzobného riadenia. Arduino je jednoduché, ale výkonné zariadenie, ktoré poskytuje podmienky vhodné pre programátorov, začiatočníkov. Pôvodne bolo navrhnuté pre umelcov a dizajnérov na vytvorenie elektronického prototypu svojich projektov [13]. Tieto nárvhy by boli schopní vytvoriť len s malou znalosťou programovania a elektroniky, bez toho, aby sa do toho dostali príliš hlboko [11].

Existuje široká škála Arduino dosiek, ale táto práca sa venuje len hardvérovej verzií rozdelenia pinov R3, kde patrí aj Arduino Uno. Arduino Uno je mikroprocesorová doska na báze Atmel ATmega328, čo je vysoko výkonný 8-bitový AVR mikrokontrolér. [11]. Rozšírením Arduina sú takzvané "štítov", ktoré sa montujú na vrch Arduina. Bývajú obľúbenými učebnými pomôckami pre študentov, ale aj pre odborníkov.

Pozitívnu charakteristickou črtou Arduina je Open-Source zdrojový kód, čo znamená, že informácie môžu byť voľne šíriteľné podľa potreby užívateľov. Ďalšími výhodami je nízka cenová dostupnosť a Arduino IDE, čo je programovacie prostredie pre Arduino softvér [29]. Čo si veľa používateľov neuvedomuje je, že usporiadanie fyzickej a elektrickej časti na Arduino Uno je tak populárne, že ho využívajú aj iné firmy pre svoje hardvéry, pričom ich parametre dosiek sú dosť odlišné. To umožňuje zostrojovať štítov pre rôzne typy prototypových dosiek bez väčších komplikácií s tým, že je možné pre ne použiť rovnaké programové aplikácie.

V Arduine IDE vývojovom prostredí sa využíva modifikovaná kombinácia programovacích jazykov C a C++. Dokonca aj niektoré profesionálne programové vývojové prostredia

teraz ponúkajú komplikácia dialektu Arduino C / C ++ priamo [29]. Arduino IDE beží na rôznych platformách ako napríklad Windows, Mac OS X a Linux.

Arduino neoslovilo len svet domácich kutilov alebo amatérskych tvorcov. Vyššie akademické inštitúcie vyriešili viaceré svoje problémy týkajúce sa nedostatkov zdrojov na nakúpenie laboratórneho vybavenia. Prístroje na meranie a pokusy sú v dnešnej dobe veľmi nákladné a študenti pracujúci na experimentoch sú nútení ho vykonávať len na akademickej pôde pod prísnym dohľadom vyučujúceho.

1.2 Projekty vo svete

Prototypová doska sama o sebe rozširuje možnosti štandardnej informatiky a programovania. Jeho skutočná hodnota sa ale ukazuje, keď je spolu s prototypovou doskou použitý aj externý hardware.

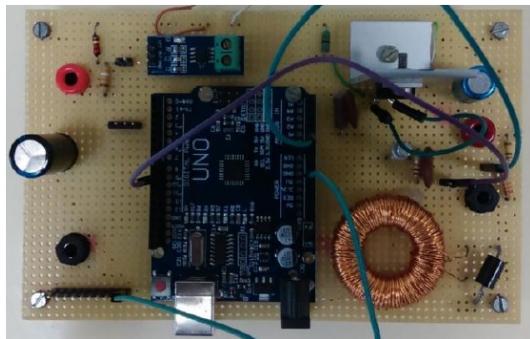
Toto si uvedomili aj výskumníci zo Stanfordskej Univerzity a z Texas Instruments [8], ktorí spolu vytvorili projekt Lab-In-A-Box. Cieľom projektu Lab-In-A-Box je zlepšenie výučby spracovania signálov a analógovej elektroniky vo vnútri aj mimo laboratórnej učebne.

Na dosiahnutie tohto cieľa vytvorili dva štítky spolu s príslušným softvérom. Prvá z dosiek — Analógový Shield — poskytuje ADC a DAC. Používa sa v niekoľkých príkladoch, ktoré učia koncepty spracovania signálov v praktickom nastavení. Druhá doska — DSP Shield — spája plne funkčný procesor s digitálnym signálom a poskytuje širokú škálu zložitosti vo vyučovaní signálových aplikácií. Projekt umožňuje študentom použiť lacné nástroje pre projekty počas ich štúdia s vysokou vernosťou, s ktorými môžu pracovať aj mimo laboratória.

Ahmad Adamu Galadima z Nigerijskej Tureckej Nilskej Univerzity [11] zase predvieďol využitie Arduino štítu na záznamník údajov. Jedná sa o záznamník údajov, ktorý by zaznamenával pH, oxidačno-redukčný potenciál (ORP) a teplotu, čo sú najbežnejšie tri parametre, ktoré sa skúmajú vo vodných systémoch a odpadových vodách. Projekt má jednoduchú implementáciu: posudzovanie vhodnosti vody pre živočíchy žijúcej v nej a vplyv zmeny prostredia na ich fungovanie. Ale samozrejme má aj iné využitia ako napríklad: monitorovanie jazier a riek a následné využitie v laboratórnom prostredí. Tento návrh shieldu je považovaný za veľmi praktický a užitočný aj keď veľká nevýhoda može byť v náročnosti a cene potrebného dopaľujúceho vybavenia ako: pH sonda, ORB sonda, pH ORP obvody, konektory BNC, teplotný senzor a mnohé iné.

Príspevok [15] predkladá prevodník napäťia riadený mikroprocesorom integrovaným na doske Arduino Uno. Riadenie je implementované pomocou PI regulátora zabudovaného na doske Arduino Uno. Funkcia „Open-to-output transfer“ je získaná z meranej krokovej odozvy. PI regulátor a prenosové funkcie spätnoväzbového oddelovača sú vytvorené tak, aby sa dosiahol požadovaný zsílenie slučky. Ako overenie analýzy sa porovnajú vstupné napäťové odozvy regulovaného a neregulovaného meniča a identifikujú sa zlepšenia. Menič je spravený vo forme obvodu pomocou prototypovej bezspájkovej dosky, ako môžeme vidieť na Obr. 1.1, avšak nie je možné ho priamo vsunúť do pinov Arduina, čo nepridáva na jeho praktickosti a jednoduchosti.

Na strednej škole v Brazílii uskutočnili vyučujúci lacné experimenty týkajúce sa obsahu učiva 2. ročníka [24]. Fyzikálne pokusy zahŕňajúce vedomosti z termodynamiky, vlnenia



Obr. 1.1: Celkový potrebný hardvér na prototyp meniča [15].

a optiky mali veľký úspech medzi študentami strednej školy. Až 94 % vyjadrilo pozitívny ohlas v elektronickom formulári, ktorý mali študenti vyplniť po 11 stretnutiach, na ktorých používali Arduino [24]. Uviedli, že pokusy boli zaujímavé alebo veľmi zaujímavé, čo však nezmenilo ich pohľad na zložitosť predmetu.

Vyvinúť modul automatického zaťaženia elektrickej stirete pre výučbu a výskum sa podarilo aj Univerzite Elizade v Nigérii [23]. Modul využíva mikrokontrolér Arduino v kombinácii so spínacím obvodom na implementáciu plánu odľahčenia záťaže v štyroch laboratórnych nasimulovaných mestských častiach (Obr. 1.2). Cieľom je zabezpečiť optimálne využitie dostupného výkonu 7,5 MW oproti spotrebe energie 10 MW. Naprogramovaný mikrokontrolér sa používa na automatické vykonávanie spínacích operácií, a to bez ľudského faktoru, čím bola úplne odstránená ľudská chyba [23]. Vyvinutý modul je užitočný ako učebná pomôcka na ilustráciu automatizácie energetických systémov medzi vysokoškolákmi, pričom využíva lacné a lokálne dostupné materiály.



Obr. 1.2: Simulácia energetickej spotreby štyroch mestských častí pomocou Arduina [23].

Nedostatok personálu a finančných prostriedkov najmä v dôsledku hospodárskej krízy, donutilo Grécku univerzitu ATEI v Solíne [10] vyvinúť laboratórne súpravy, aby slúžili potrebám vysokoškolských a postgraduálnych kurzov. C.Y.Foulis a spol. [10] vrvavia: „Nevyhnutná bola drastická reforma laboratórnych kurzov, ktoré sú teraz založené na lacných súpravách s domácimi spotrebičmi vybavenými technológiou Arduino.“

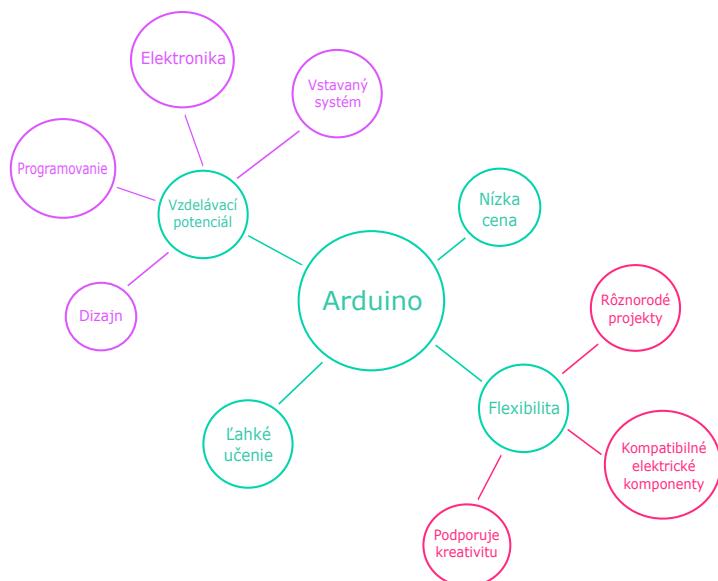
S celkovými nákladmi 200 EUR vytvorili 10 kompletných laboratórnych súprav, ktoré umožňujú vykonať rôzne experimenty, zahŕňajúce laboratórne skupiny pozostávajúce až z 20-30 študentov. Experimenty zahŕňajú ovládanie jasu LED, reguláciu teploty, reguláciu otáčok a polohy servomotora, ako aj riadenie polohy kyvadla.

Súprava bola navrhnutá tak, aby podporovala oblasť číslicového riadenia, v ktorom sa

študenti naučia diskretizovať typické analógové regulátory a naprogramovať ich vo forme diferenčných rovníc. Taktiež sa naučia programovať praktické PID regulátory s extra funkciami.

V Českej republike na Juhočeskej univerzite v Českých Budejoviciach navrhli koncept vzdelávania na gymnáziách a stredných odborných školách s využitím platformy Arduino. Projekt bol ešte len v testovacej fáze, no ukázalo sa, že ho oceňujú najmä študenti gymnázií. Na stredných odborných školách, kde majú študenti už znalosti z elektrotechniky, považovali experimenty s Arduinom Unom za jednoduché [22].

Cieľom príspevku [22], bolo poukázať na to, ako používať mikroprocesor Arduino v učebných osnovách pre študentov stredných a vysokých škôl s cieľom získať a podporiť logické myšlenie. Taktiež na Obr. 1.3, stručne popísali výhody Arduina vo vzdelávacom systéme, kde medzi hlavné uviedli: nízka cena, flexibilita, jednoduchá manipulácia, rozmanitosť projektov a vzdelávací potenciál [22].



Obr. 1.3: Výhody Arduina v edukačnom systéme [22].

Na jednej z univerzít na Taiwane využili Arduino ako digitálny ovládač pre cvičenia z teórie automatického riadenia. Systém s názvom "SOAC" [16] bol skúšaný na digitálnom osciloskope, kde skokovým signálom merali dobu nábehu, čas oneskorenia, maximálny kmit a čas ustálenia. Číslicový regulátor bol implementovaný na Arduino UNO. Prezentujú dva demonštratívne príklady praktických experimentov, jeden na PID riadenie a druhý na Fuzzy riadenie. Navrhnutý učebný plán pre teóriu automatického riadenia môže posilniť praktické skúsenosti študentov, študujúcich na inžinierskom stupni. V príspevku [16] navrhujú inovatívne nastavenie laboratórneho systému a návrh učebných osnov.

Aj na Slovensku sa Arduino pomaly, ale isto zaraďuje do vzdelávania. Celosvetovo patrí do učebných osnov 13 univerzít, z toho až 3 sú na Slovensku [14].

Vzdelávacia schéma z článku [14] bola úspešne zaradená do vzdelávacieho procesu Katedry počítačov a informatiky Technickej univerzity v Košiciach. Ako ukazujú výsledky

dotazníka, pripravené vyučovacie hodiny boli úspešné z hľadiska študijných materiálov, náročnosti úloh, použitého hardvéru a zvolenej témy. Vzdelávací systém by sa mal vedieť prispôsobiť rastúcemu záujmu o mikrokontroléry a technológie Arduino. Aj za týmto účelom sa v priebehu akademického roka vyvíja mnoho bakalárskych a diplomových prác.

1.3 Projekt AutomationShield

Bohužiaľ u všetkých uvedených v Kap. 1.2 a väčšine zrealizovaných projektov sa jednalo len o amatérsky prevedenie hardvéru. Improvizované mechanické komponenty a konštrukcie sú vyrobené prevažne z dostupných materiálov v danej oblasti s cieľom, aby náklady boli čo najnižšie. Detaily o konštrukcií nie sú dokumentované alebo len nedostatočne.

Myšlienka všetkých experimentov je použiteľná pre viaceré projekty do budúcnosti, ale čo sa týka hardvéru sa to povedať nedá. Kvôli nepostačujúcim informáciám o presnom zhodení je teda málo pravdepodobné, že by sa rovnaký projekt dostal do učebných osnov iných inštitúcií, ako tam kde vznikol.

Veľmi negatívny vplyv individuálnych pedagogických projektov s využitím Arduina vzniká pri riešení problémov pri programovaní. Keďže medzi sebou vzdelávacie inštitúcie venujúce sa jednej oblasti nekomunikujú, riešia obvykle tie isté problémy, ale každý na svojom bojovom poli. Výrazne to znižuje efektivitu práce.

S cieľom zlepšiť vyššie spomenuté nedostatky prišiel projekt AutomationShield. Je to Open-Source hardvérový a softvérový projekt zameraný na nástroje pre výučbu automatizácie a mechatroniky [12].

Jadrom projektu sú rozšírenia dosiek, známe ako Arduino štíty. Ich primárny účel je ako experimentálne zariadenia na spätnú väzbu a zároveň poskytnú aplikačné programátorské rozhranie (API). Spolu so súborom CAD na výrobu PCB je poskytnutý aj zoznam požadovaných komponentov, súbory potrebné pre 3D tlač a knižnice s metódami na ovládanie štítu. Návrh sa vždy snaží vyhýbať neobvyklým a drahým komponentom [12].

2 Tvorba hardvéru

Vytvorenie experimentálneho zariadenia na riadenie statickej deformácie votknutého nosníka sa skladá z dvoch základných problematík a to tvorba hardvéru a softvéru. V tejto kapitole je popísaný postup pri narhovaní vhodných komponentov a ich tvarov, vývoj prototypov a návrh elektrického obvodu.

Hlavné dva prvky na prototype sú servo motor a nosník, na ktorom je upevnený flexi snímač. Rameno serva má za úlohu ohýbať kovovým nosníkom. Pomocou flexi snímača sa zistí veľkosť ohybu. Cieľom je pomocou servo motora nastaviť požadované hodnoty ohybov nosníka.

2.1 Popis a parametre použitých komponentov

Cieľom pri výbere komponentov boli hlavne ich technické parametre. Vytvorenie obvodu vyžadovalo aby boli všetky súčiastky schopné byť napájané z Arduina. Druhým faktorom bola ich veľkosť. Preferované boli SMD (Surface Mount Devices, v preklade zariadenia na povrchovú montáž) komponenty, aby sa zmestili na prototypovú dosku veľkosti Arduina Una.

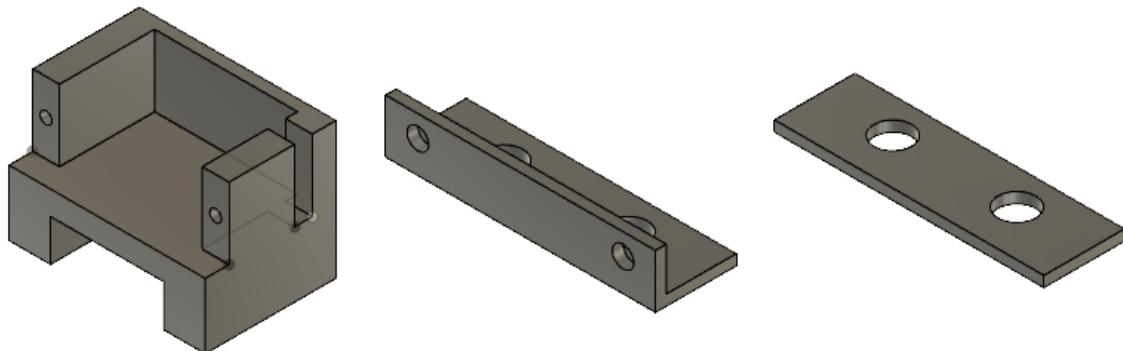
2.1.1 Prototypová doska pre Arduino UNO

Rozmery PBC (Printed Circuit Board, v preklade vytlačená obvodová doska) dosky sú $68,58 \times 53,34$ mm a hrúbka 1,6 mm. Materiál tvorí sklenené vlákna FR-4. Všetky otvory sú pokovované, cez hornú a spodnú stranu sú medené vrstvy. Mriežku podložky tvoria diery s priemerom 0,9 mm, ktoré sú od seba vzdialené 2,54 mm [4]. Prototypová doska vytvorila základ, na ktorý sa umiestnili všetky potrebné súčiastky.

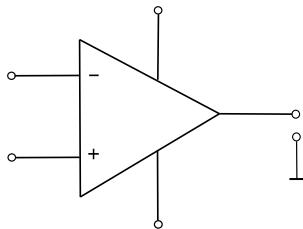
2.1.2 Komponenty vyrobené pomocou 3D tlače

Komponenty boli vytlačené technológiou FDM (Fused Deposition Modeling, teda generatívne konštruovanie), ktorá spolupracuje so špecializovanými 3D tlačiarňami a termoplastmi na výrobu pevných, odolných a rozmerovo stabilných časťí [27]. Výhody tejto metódy sú [27]:

- technológia je čistá, ľahko použiteľná a pohodlná pre kancelárie,
- podporované termoplasty sú mechanicky a environmentálne stabilné,



Obr. 2.1: Komponenty vytlačené pomocou 3D tlače.



Obr. 2.2: Schematická značka operačného zosilňovača.

- komplexné geometrie a dutiny, ktoré by inak boli problematické, sa stávajú praktickejšími s technológiou FDM.

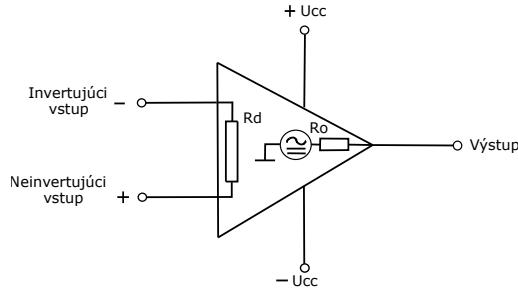
PET-G materiál je upravená verzia materiálu PET (polyetylentereftalát). "G" znamená modifikovaný glykol, ktorý sa pridáva počas polymerizácie. Výsledkom je vlákno, ktoré je menej krehké, ľahšie sa používa a je vysoko odolný proti nárazu, čím sa líši od PET. V chémii sa tento materiál označuje ako polyethylentereftalátko-1,4-cyklohexylendimethylenetereftalán. PET-G vlákno kombinuje vlastnosti materiálov ABS (Akrylonitrilbutadiéns-tyrén) a PLA (Polylaktid), teda je silnejší, odolnejší voči teplotám, ale zároveň tlač s týmto materiálom nie je komplikovaná. Riziko skrútenia alebo zmrštenia je nízke a veľmi priažnivá vlastnosť je, že materiál je možno recyklovať [18].

Výtlačky boli použité na upevnenie serva a nosníka pre snímač ohybnosti (Obr. 2.1). Presný popis a tvar súčiastok je popísaný v Kap. 2.2.1 až 2.2.2

2.1.3 Operačný zosilňovač

Operačný zosilňovač (OZ) so schematickou značkou na Obr. 2.2, je rozdielový zosilňovač s veľmi vysokým zosilnením pre jednosmerné aj striedavé signály realizovaný v integrovanej forme [3]. Základom jeho vnútornej štruktúry je diferenciálny jednosmerný zosilňovač. Ide o univerzálny obvod so širokým spektrom využitia. OZ môže mať vstup aj výstup symetrický alebo nesymetrický [3]. Vnútorné rozhranie operačného zosilňovača na Obr. 2.3 obsahuje:

- neinvertujúci vstup – nespôsobuje otočenie fázy výstupného signálu



Obr. 2.3: Vnútorné rozhranie operačného zosilňovača.

- invertujúci vstup – spôsobuje otočenie fázy
- výstup
- napájanie kladné +UCC a záporné -UCC

Zariadenie LM358ADR je verzia novej generácie štandardného zariadenia LM358. Má široký rozsah napájania od 3 V do 36 V pri prúde $300 \mu A$ so šírkou pásma 1,2 MHz. Obsahuje dva nezávislé vysokonapäťové operačné zosilňovače s kompenzáciou kmitočtu, ktoré sú určené na prevádzku z jedného zdroja na širokú škálu napätí. Operačný zosilňovač zjednodušuje návrh obvodov s vylepšenými funkciami, ako je stabilita a nižší pokojový prúd $300 \mu A$. LM358ADR môže byť prevádzkované priamo zo štandardného 5 V napájania používanej v digitálnych systémoch a jednoducho môže poskytnúť požadované rozhranie elektroniky bez ďalších napájacích zdrojov [31].

2.1.4 Rezistory

Rezistory (Obr. 2.4) sú pasívne prvky, ktoré sa používajú sa vo všetkých elektrických obvodoch. Základnou vlastnosťou rezistora je jeho odpor [28]. Rezistory používané v elektrotechnike sú vhodne upravené materiály, ktoré kladú pretekajúcemu prídu odpor. Veľkosť pretekajúceho prídu je nepriamoúmerná veľkosť odporu. Pretekaním prídu rezistorom vzniká na ňom úbytok napäcia priamoúmerný veľkosť pretekajúceho prídu a hodnote rezistora. Rezistory sa používajú na obmedzenie prídu a napäcia, prípadne na ich reguláciu [28].

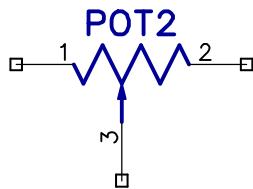
Použité rezistory sú povrchové komponenty a ich rozmery sú veľmi malé. Dĺžka 2 mm, výška 0,55 mm a šírka 1,25 mm. Rezistor sa prevádzkuje pri minimálnej teplote -55°C a maximálnej $+155^{\circ}\text{C}$ s toleranciou 5 %.



Obr. 2.4: Schematická značka rezistora.

2.1.5 Potenciometer

Potenciometer je regulovateľný delič napäťia, ktorý odoberá napätie len z určitej časti odporu, ktorým preteká elektrický prúd. Ide o variabilný rezistor s pripojeným bežcom, ktorý je v kontakte s odporovou vrstvou a je vyvedený na samostatný vývod (Obr. 2.5) [2]. Je to jednoduché zariadenie, poskytujúce variabilný odpor, ktorý môžeme čítať do dosky Arduino ako analógovú hodnotu. Potenciometer obsahuje tri kolíky. Prvý z nich je pripojený k zemi, druhý do 5 V a tretí ide do analógového výstupu aby sme mohli čítať hodnotu potenciometra. Otáčaním hriadeľa potenciometra meníme odpor, ktorý je pripojený k stredovému kolíku potenciometra. To mení elektrický potenciál kolíka od 5 V, čo dáva iný analógový výstup [2]. V prototype nebol potenciometer nevyhnutným komponentom, použili sme ho z dôvodu širšej možnosti programovania.



Obr. 2.5: Schematická značka potenciometra.

2.1.6 Nosník

Nosník je vyrobený z nerezovej ocele AISI 301. Je to austenitický nehrdzavejúci materiál vhodný na tvarovanie. Jeho označenie je 301 a číslo UNS je S30100. Je bežne dostupný s dobrou odolnosťou voči korózii a zvýšeným obsahom uhlíka, aby sa umožnilo opracovanie za studena (Tab. 2.1). Vyskytuje sa vo viacerých formách: 1/4 tvrdej, 1/2 tvrdej, 3/4 tvrdej, plno tvrdej alebo extra tvrdej forme. V žíhanom stave sa ľahko formuje. Pri tvárniciach prácach, pri ktorých dochádza k deformácii materiálu v temperovanom stave, sa musí postupovať opatrne, pretože môže dôjsť k praskaniu a roztrhnutiu. Zvárať sa môže akýmkoľvek známym spôsobom. Materiál nereaguje na kalenie [17] .

Z dôvodu, že materiál má špecifické fyzikálne vlastnosti, viď. Tab. 2.2, nie je možné ho prevítať ani prepíliť bežnými prostriedkami. Na jeho spracovanie bola použitá technológia laserového rezania, ktorej základom je laser slúžiaci na zosilnenie elektromagnetického žiarenia [20]. Žiarenie má v každom svojom bode rovnakú vlnovú dĺžku, vďaka čomu je možné vysielať ho v tenkom lúči.

Laser vytvorí lúč a priviedie ho na miesto rezu. Začne sa zahrievať materiál a následne sa odparuje, topí, prípadne zhorí. Pri procese sa používa pracovný plyn, ktorý odstraňuje zvyšky roztaveného plynu tým vznikne veľmi kvalitný povrch rezu. Laser je vhodný na delenie materiálov s hrúbkou do 10–15 mm. Nevýhodou technológie rezania laserom je vyššia ekonomická a prevádzková náročnosť [20].

Nosník musel splňať požadované vlastnosti. V prvom rade musel byť dostatočne pružný aby ho servo motor dokázal ohnúť viď. Kap 2.1.7 a v druhom rade nemohol stratiť pri deformáciach svoj pôvodný tvar. Tieto dva dôvody, ale samozrejme aj iné nám pomohli rozhodnúť sa, že použijeme nosníky práve z materiálu AISI 301. Nosník sa tvarovo aj

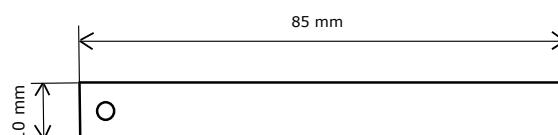
Tabuľka 2.1: Chemické zloženie ocele AISI 301[19]

Prvok	Percentuálny podiel
Uhlík	0,15%
Chróm	16 – 18 %
Ocel'	70 – 75 %
Mangán	2 %
Nikel	6 – 8 %
Fosfor	0,045 %
Kremík	1 %
Síra	0,03 %

Tabuľka 2.2: Fyzikálne vlastnosti ocele AISI 301[19]

Vlastnosť	Hodnota
Hustota	4,65 kg/m ³
Špecifická hmotnosť	7,8
Špecifické teplo	0,5024 KJ/kg . K
Elektrická rezistivita	432 ($\mu\Omega$)/cm
Modul pružnosti pri ťahu	28

rozmernovo menil s vývojom prototypov, no konečná verzia mala podobu jednoduchého obdĺžnika s dĺžkou 85 mm a šírkou 10 mm, viď. Obr. 2.6.



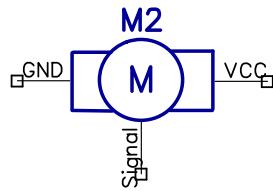
Obr. 2.6: Rozmery nosníka.

2.1.7 Servo motor

Servo motor je otočný pohon alebo lineárny pohon, ktorý umožňuje presné riadenie uhlovej alebo lineárnej polohy, rýchlosťi a zrýchlenia [1]. Pozostáva z vhodného motora pripojeného k snímaču polohy (Obr. 2.7). Štandardné hobby servo sa môže otáčať približne o 180 stupňov a to 90 stupňov v každom smere. Prevedenie "micro" funguje rovnako ako štandardné druhy, ale je menšie, čo mu dáva výhodu, že sa zmestí aj na malé miesta, ale samozrejme, nie je tak silné. Dobre pracuje s motorovým štítom pre Arduino alebo len prepojením s knižnicou Servo [1].

Vlastnosti použitého servomotoru:

- Rozmery: $23 \times 11 \times 29$ mm
- Napätie: 3 V až 6 V DC
- Hmotnosť: 9 g
- Rýchlosť: 0,12 s / 60° (pri 4,8 V)
- Krútiaci moment: 0,16 Nm
- Pracovná teplota: $-30^\circ\text{C} - 60^\circ\text{C}$

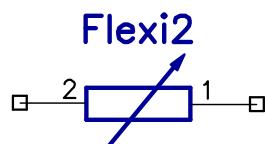


Obr. 2.7: Schematická značka servo motora.

2.1.8 Snímač ohybnosti - flexi senzor

Snímač ohybnosti je snímač, ktorý meria veľkosť vychýlenia alebo ohybu. Zvyčajne je senzor prilepený k povrchu a odpor snímacieho prvku sa mení ohýbaním povrchu. Odpor je priamo úmerný množstvu ohybu a často sa nazýva aj flexibilný potenciometer [26]. Do obvodu ho zapájame ako deličku napätie, čomu zodpovedá aj jeho schematická značka, Obr. 2.8.

Životnosť flexi senzora je viac ako 1 milión cyklov, čo vyhovuje pri tvorbe didaktickej pomôcky, ktorú budú dennodenne používať študenti. Celková dĺžka komponentu je 112,24 mm z toho senzor tvorí dĺžku 95,25 mm. Šírka flexi snímača je 6,35 mm a hrúbka len 0,43 mm. Snímač pracuje správne v teplotnom rozsahu od -35°C až $+ 80^\circ\text{C}$. Výrobca odporúča použiť s týmto komponentom operačný zosilňovač LM358 alebo LM324 [26].



Obr. 2.8: Schematická značka flexi snímača.

2.1.9 Kondenzátor

Kondenzátor, ktorého schematickú značku je možné viedieť na Obr. 2.9, je pasívna elektronická súčiastka, ktorá uchováva energiu vo forme elektrostatického poľa [34]. Vo svojej najjednoduchšej forme sa kondenzátor skladá z dvoch vodivých dosiek oddelených izolačným materiálom nazývaným dielektrikum. Kapacitancia kondenzátora je priamo úmerná povrchovému obsahu dosiek a je nepriamo úmerná vzdialenosť medzi doskami. Taktiež závisí od dielektrickej konštanty látky oddeľujúcej dosky [34]. Vlastnosti použitého kondenzátora [21]: Tantálový kondenzátor má kapacitanciu $100 \mu F$. Toleranciu má 20 % a správne funguje v teplotnom intervale od $-55^{\circ}C$ do $+125^{\circ}C$.



Obr. 2.9: Schematická značka kondenzátora.

2.1.10 Dióda

Dióda je polovodičové zariadenie, ktoré funguje ako jednosmerný prepínač prúdu. Umožňuje prechod elektrického prúdu len v jednom smere a silno obmedzuje prúd v opačnom [30].

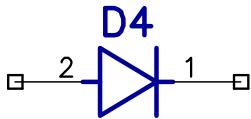
Diódy sú známe aj pod pojmom usmerňovače, pretože menia striedavý prúd (AC - Alternating Current) na jednosmerný prúd (DC - Direct Current). Diódy majú polaritu určenú anódou a katódou. Väčšina diód umožňuje prúdenie prúdu len vtedy, keď je na anódu privedené kladné napätie [9].

Vlastnosti použitého usmerňovača [33]: Použili sme usmernovač formou prepínacej diódy, Obr. 2.10, s rozmermi výška 2,74 mm, dĺžka 4,75 mm a šírka 2,9 mm. Je tak isto v prevedení SMD. Výrobca odporúča pracovať s usmerňovačom pri minimálnej teplote $-65^{\circ}C$ a maximálnej $+175^{\circ}C$. Ďalšie vlastnosti použitého usmerňovača [33]:

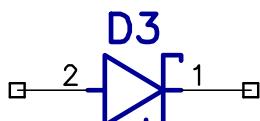
- Vrcholové reverzné napätie: 200 V
- I_f - prúd v prepúšťacom smere: 1 A
- Čas obnovenia: 150 ns
- V_f - napätie v prepúšťacom smere: 1,3 V
- Vrcholové reverzné napätie: 200 V

2.1.11 Zenerova dióda

Zenerova dióda je špeciálny typ diódy (Obr. 2.11). Používa sa na výrobu stabilizovaného napäťa s nízkym vlnením aj pri meniacich sa podmienkach záťažového prúdu [25]. V zenerovej dióde vzniká presne definovaný úbytok napäťa. Často sa využíva na vytvorenie referenčného napäťa alebo v regulátoroch napäťa.



Obr. 2.10: Schematická značka diódy.



Obr. 2.11: Schematická značka zenerovej diódy.

2.2 Vývoj prototypov

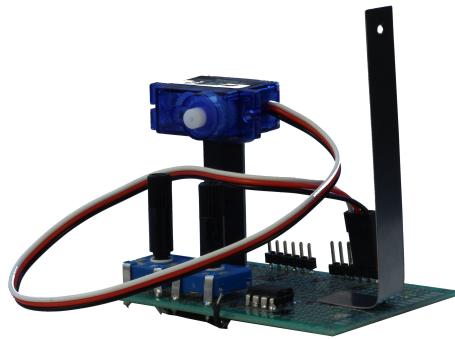
Skonštruovaniu finálneho experimentálneho zariadenia na riadenie statickej deformácie votknutého nosníka s pracovným názvom TugShield, predchádzala výroba troch prototypov. S každým ďalším prototypom prišli nové vylepšenia, ktoré odstraňovali nedostatky na tom predchádzajúcim. Nemenilo sa len uloženie komponentov na prototypovú dosku ale aj ich tvar, rozmery, prípadne boli pridané nové potrebné komponenty.

2.2.1 Prvý návrh

Návrh konštrukcie TugShieldu prinášalo viaceré alternatívy. Najdlhšia súčiastka na prototype je flexi snímač, ktorý ma dĺžku 7 cm a je pripojený na kovový nosník, na ktorom musí byť upevnený motúz. Takže celková dĺžka celého tohto komponentu je okolo 7,5 cm. Predstavovalo to dve možnosti uloženia na štít, a to vertikálne a horizontálne. Pri horizontálnom uložení by bola vzdialenosť medzi servo motorom a flexi snímačom približne 4 cm. Pri vertikálnom by bola cca 7 cm, z čoho som usúdila, že čím väčšia vzdialenosť tým to bude poskytovať lepšie podmienky a v konečnom dôsledku aj lepšie výsledky merania. Taktiež z hľadiska manipulácie a upevnenia bola vertikálna možnosť umiestnenia flexi člena lepšia voľba.

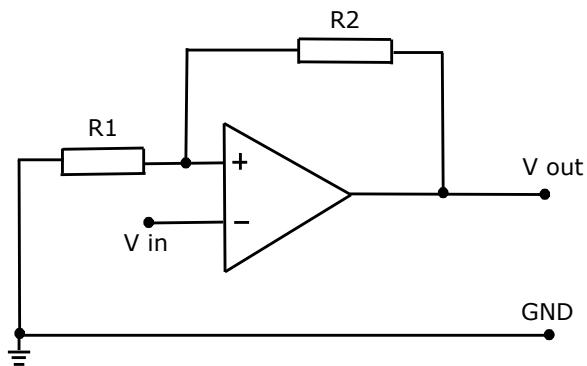
V prvom prototype bol snímač ohybnosti nalepený na kovom nosníku v tvare L. Strana, na ktorej bol flexi senzor upevnený, mala rozmery 100×10 mm a kratšia časť L nosníka mala rozmery 40×10 mm. Táto strana bola upevnená na prototypovú dosku Arduino Uno a vytvárala tak vertikálnu polohu snímača.

Ďalšou veľmi dôležitou otázkou bola poloha servo motora z hľadiska jeho výšky voči flexi snímača. Keby sa servo nachádzalo príliš nízko a blízko pri senzore, ohyb by sa uskutočňoval prevažne len v hornej časti nosníka, pričom flexi je umiestnený 7 mm od horného okraja smerom dolu. Ohyb by nebol spôsobený v takom rozsahu. Preto sa servo umiestnilo na plastový nosník s výškou 4 cm vo vzdialosti od snímača ohybu 6,5 cm, čo bolo najviac, ako to umožňovala konštrukcia prototypovej dosky. Servo motor bol prilepený na nosník, a tak isto nosník na dosku, avšak táto metóda spájania neboli najideálnejšia z hľadiska stability a sily spoja, ale jednalo sa len o prvotné zostrojenie prototypu (vid. Obr. 2.12).



Obr. 2.12: Prvý prototyp.

Na zosilnenie signálu, ktorý vytváral flexi snímač, bol použitý operačný zosilňovač v konfigurácii jednoduchého neinvertujúceho zosilovača. Dôvod, prečo bol vybratý práve tento zosilňovač bol, že vykonáva funkciu zosilňujúceho impedančného meniča s veľkým vstupným a malým výstupným odporom. Teda zachová pôvodnú charakteristiku signálu, len ho zväčší. Schému zapojenia neinvertujúceho operačného zosilňovača je možné vidieť na Obr. 2.13. Pri zapojení operačného zosilňovača som použila namiesto rezistorov so



Obr. 2.13: Zapojenie neinvertujúceho operačného zosilňovača.

stálymi hodnotami odporu potenciometre, a tým som pri meraní zistila, v akej kombinácii hodnôt bude moje zosilnenie najefektívnejšie.

Prototypová doska mala rozmery 50×70 mm a nelemovala presne tvar Arduino Uno. Nenachádzali sa tu kolíky vedené do pinov na Arduine, všetky elektrické výstupy z obvodu boli smerované hore a nie dole, takže tento prvý prototyp nebolo ešte možné zapojiť priamo na Arduino Uno. Na spojazdnenie musela byť použitá bezpájkové kontaktné pole.

2.2.2 Druhý návrh

Druhý návrh bol už urobený v prevedení štítu, takže sa dal priamo umiestniť na Arduino. Základná doska mala rozmery $68,58 \times 53,34$ mm s hrúbkou 1,6 mm. Na krajoch boli predvŕtané diery na kolíky, ktoré presne viedli do pinov na Arduine. Spojazdnenie tohto prototypu bolo o to jednoduchšie.

Veľkou zmenou v porovnaní s prvým prototypom bolo upevnenie kovového nosníka s flexi snímačom. Snímač už nebol nalepený na kovovom L-ku, ale bol to len jednoduchý nosník s rozmermi 11×105 mm. Ten bol uchytený zverným spojom, ktorý predstavovali dva komponenty vytlačené na 3D tlačiarni. Obidva boli v tvare L, zrkadlovo otočené. Dlhšou časťou, ktorá mala rozmery 50×10 mm sa L-ká pritlačili k sebe a medzi ne sa umiestnil nosník. Kratšou stranou sa L-ká pripevnili k základnej doske. Spojovacia



Obr. 2.14: Druhý prototyp.

technika už nebola v podobe lepidla, ale do vytlačených komponentov sa navrtali diery. Na dolnej podstave mali priemer 4 mm a boli umiestnené symetricky od okraja 6 mm. Na hornej podstave sa diery vyvrtali tiež, a to na uchytenie nosníka s flexi snímačom. Tieto diery mali priemer 6,5 mm a od okraja boli umiestnené 14 mm, taktiež symetricky. Kovový nosník mal šírku 10 mm a snaha bola umiestniť diery čo najbližšie k nemu, aby bol upevnený čo najsilnejšie. Do dier sa potom vložili skrutky s vonkajším priemerom 6 mm, a tým sa vytvorilo zovretie. Rozloženie vyvŕtaných dier bolo premyslené tak, aby si skrutky navzájom nezavadzali a spoj sa dal kedykoľvek povoliť, aby sa mohol vymeniť kovový nosník za nosník s iným materiálom (viď Obr. 2.14).

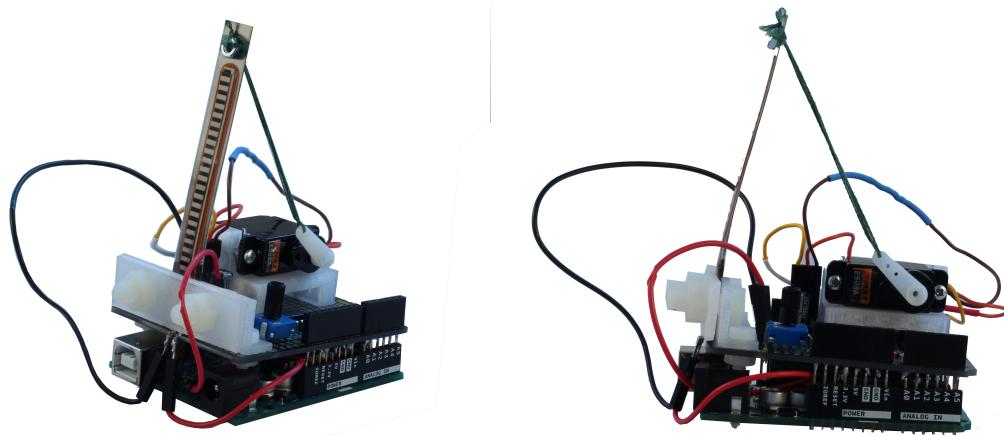
Ďalšie vylepšenie prišlo zo strany uchytenia servo motora. Cieľom bolo dosiahnuť čo najpevnejšie uchytenie, čo zahŕňalo využitie dier, ktoré sa nachádzali na krídlach serva. Servo bolo umiestnené vertikálne, aby rameno mohlo čo najviac ohýbať kovový nosník a následne bolo vložené do výtačku v tvare U, kde sa priskrutkovalo, cez diery. Celý tvar U bol vyzdvihnutý do potrebnej výšky, aby rameno servo motora mohlo vytvoriť dráhu celej kružnice. Na bočnej strane bola na U vytvorená medzera pre káble, ktoré viedli z motora. Tento prvok bol tak isto vytlačený v 3D tlačiarni a predstavoval najmohutnejšiu časť celého prototypu. Na spodnej časti, ktorá dosadala na základnú dosku bola vyvŕtaná diera rovnako ako aj do dosky a tieto dve časti sa priskrutkovali dokopy. Veľká nevýhoda v tomto spojení bola to, že ak by sa skrutka povolila pri sile, ktorú by vytvorilo servo ťahajúce kovový nosník s flexi snímačom, celé upevnenie servo motora by sa mohlo pootočiť, čo by malo nepriaznivý dopad na výsledky merania.

Negatívne na druhom skonštruovaní prototypu bola jeho váha. Celý štít vážil 74 g čo bolo o 41 g viacej ako prvý prototyp. Jednak vytlačené tvary boli veľmi mohutné a

zaberali veľa priestoru na doske, ale najväčšiu váhu mali kovové skrutky, ktoré boli použité na zvernom spoji.

2.2.3 Tretí návrh

Tretí prototyp sa už začínať podobať s finálnym návrhom. V tomto kroku bola snaha zminimalizovať súčiastky, ktoré boli vytlačené na 3D tlačiarni. Nie len z hľadiska hmotnosti, ale aj šetrenia materiálu, čo má dopad na ekonomickú otázku. Na dosku som chcela pridať aj nový komponent – potenciometer, ktorý zaberal určité miesto, takže výtláčky museli byť aj rozmerovo zmenšené. Z nosníka, na ktorom bolo uchytené servo, sa horná časť nezmenila, avšak dolná časť už nebola tvorená jedným kusom, ale bola nahradená dvomi menšími podlhovastými nohami, ktoré sa nachádzali na okrajoch komponentu. Tým sa vytvoril most, pod ktorým bol priestor pre ďalšie súčiastky. Rezistory sme presunuli pod tento komponent a vytvorilo sa miesto pre potenciometer. Veľká výhoda bolo eliminovanie povolenie jednej skrutky pri sile servo motora tým, že to bola doska spojená s dvomi nohami a dvomi skrutkami. Spoj sa zafixoval v dvoch bodoch, ktoré sa nemali ako povoliť.



Obr. 2.15: Tretí prototyp.

Zverný spoj bol tak isto upravený pri treťom návrhu. Snažila som sa flexi snímač umiestniť čo najviac ku kraju a tým zväčšiť vzdialenosť od servo motora. Z predošlého návrhu - dvoch zrkadlových komponentov v tvare L bol jeden upravený a to tak, že sa odstránila spodná časť, ktorá bola pripojená o dosku. Vznikol zverný spoj, ktorý bol tvorený jedným L-kom a platničkou, ktorá doliehala na jeho širšiu časť. V obidvoch komponentoch boli symetrické diery na skrutky. Tým sa dokázal upevniť nosník so snímačom ohybnosti a celková šírka sa zmenšila cca. o polovicu. Pred tým bol snímač ohybu vzdialenosť od kraja 13 mm a po týchto úpravách len 3 mm.

Kovové skrutky predstavovali väčšinu váhy prototypu číslo dva, takže boli vymenené za plastové, čím sa znížila váha na 44 g. Zmenšila sa aj dĺžka skurtiek, ktoré presahovali cez základovú dosku.

Návrh zosilnenia pomocou operačného zosilňovača

Samotný signál, ktorý vychádzal z flexi snímača nebol dostatočne silný, (vidľ. Obr. 2.16). Rozdiely medzi krajnými polohami sa pohybovali v rozhraní od 130 do 65 úrovní pri 10 bitovom A/D prevodníku, kvôli čomu som sa rozhodla použiť neinvertujúci operačný zosilňovač. Aktívny komponent LM358 má 8 pinov na zapojenie. Súčiastka obsahuje až dva operačné zosilňovače a v obvode bol využitý iba jeden z nich. Zosilňovač sa pripojil podľa zapojenia v Kap. 2.2.1 a snažila som sa nájsť vhodný pomer odporov rezistorov. Zosilnenie sa dá vypočítať pomocou Rov. (2.1).

$$A_u = U_{vst}/U_{vst} \quad (2.1)$$

$$U_{vst} = U_{R1}$$

$$U_{vst} = I \cdot R_1$$

$$U_{vst} = U_{vst} + I \cdot R_2$$

$$U_{vst} = I \cdot R_1 + I \cdot R_2$$

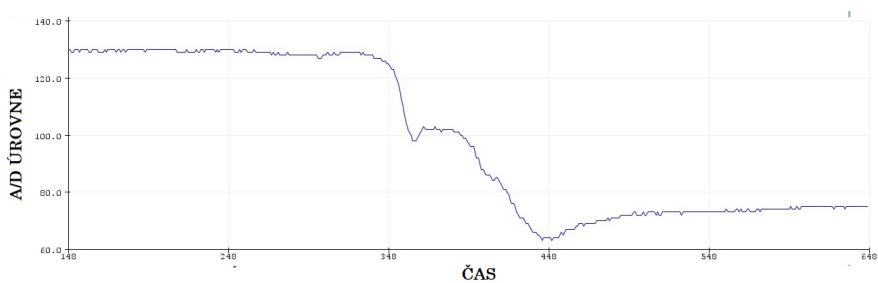
$$A_u = (I \cdot R_1 + I \cdot R_2)/I \cdot R_1$$

$$A_u = 1 + R_2/R_1 \quad (2.2)$$

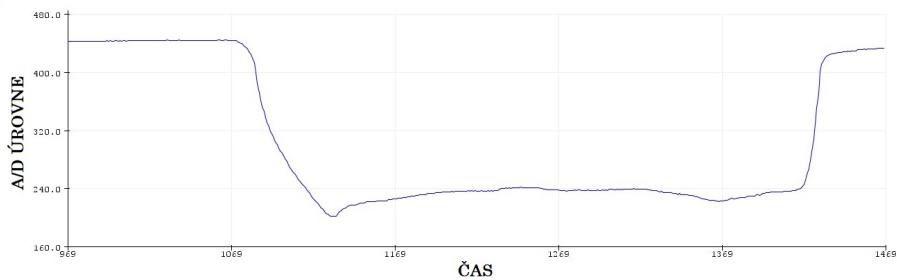
kde

- U_{vst} – vstupné napätie [V]
- A_u – konečné zosilnenie
- U_{vst} – výstupné napätie [V]
- I – prúd [A]
- R_1 – odpor na rezistore č.1 [Ω]
- R_2 – odpor na rezistore č.2 [Ω]

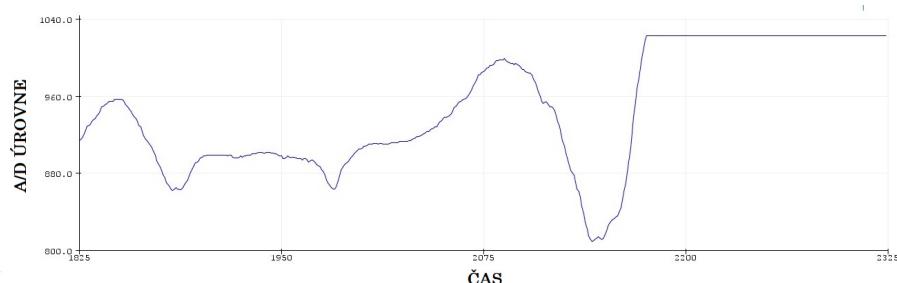
Cieľom bolo získať čo najväčšie zosilnenie s tým, aby operačný zosilňovač nesaturoval. Boli vyskúšané viaceré možnosti rezistorov. Najprv sa začalo hodnotami pre $R_2 = 11\text{ k}\Omega$ a $R_1 = 4\text{ k}\Omega$. Výstupné zosilnenie vyšlo 3,75, ktoré sa veľmi nezmenilo od nezosilneného signálu, Obr. 2.17. Ako druhú kombináciu som vyskúšala $R_2 = 16\text{ k}\Omega$ a $R_1 = 1,3\text{ k}\Omega$, pri ktorej zosilňovač začal saturovať, čo bolo nežiadúce. Na Obr. 2.18 je jasne vidieť vyhľadenú rovnú čiaru na hodnote 1023. úrovne pri 10 bitovom A/D prevodníku, čo značí o nadmernom zosilnení signálu. Celkom vhodné hodnoty pre zosilnenie boli $R_2 = 16\text{ k}\Omega$ a $R_1 = 3,3\text{ k}\Omega$ so zosilnením 5,32 vidľ. Obr. 2.19, ale na koniec sa ešte zväčšilo zosilnenie na 6,93 s hodnotami rezistorov $R_2 = 16\text{ k}\Omega$ a $R_1 = 2,7\text{ k}\Omega$, čo bolo najväčšie možné. Maximálna hodnota na Obr. 2.19 je cca 770 úrovni, s porovnaním na Obr. 2.20 je to okolo 950 úrovni. Minimálne hodnoty pre $R_1 = 3,3\text{ k}\Omega$ osilujú na hodnote 350. úrovne a pre $R_1 = 2,7\text{ k}\Omega$ na 450. úrovne.



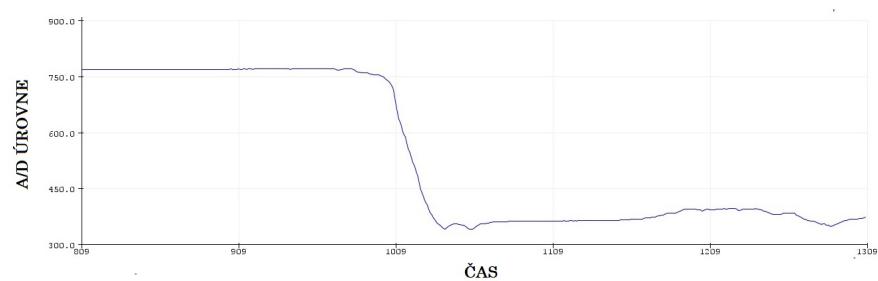
Obr. 2.16: Signál zo snímača bez zosilnenia.



Obr. 2.17: Slabé zosilnenie signálu.



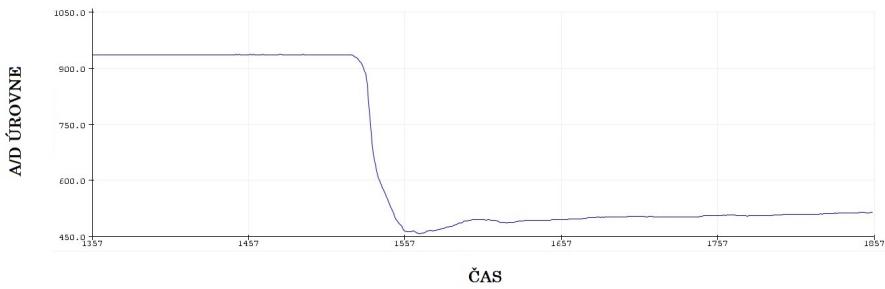
Obr. 2.18: Saturovanie signálu.



Obr. 2.19: Vhodné zosilnenie.

2.3 Vytvorenie plošnej dosky

Návrh plošnej dosky bol urobený v CAD prostredí DipTrace. Program obsahuje knižnicu často používaných komponentov, ale v tomto prípade boli použité aj komponenty, ktoré v knižnici neboli. Prvým krokom ich bolo nakresliť.



Obr. 2.20: Najvyhovujúce zosilnenie.

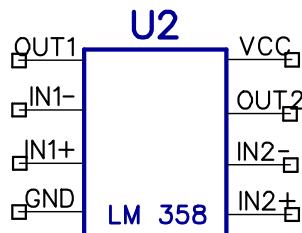
Z technického dokumentu pre operačný zosilňovač LM358ADR sa zistili odporúčané parametre plôšok (takzvané „pads“), ktoré sme nakreslili v module Pattern Editor. Tento DipTrace modul umožňuje užívateľovi spravovať knižnice vzorov a kresliť vzory s rôznymi typmi tvarov, podložiek, otvorov a rozmerov. Okrúhle, rovné, štvorcové (QFP), maticové (BGA), obdlžnikové (RQFP) štandardné šablóny poskytujú významné zrýchlenie návrhu. Vytvorenie vzoru je v podstate výber šablóny, zadanie niekoľkých nevyhnutných parametrov, nakreslenie sietotlače a spustenie automatického prečíslovania podložiek. Pre neštandardné vzory možno vytvoriť vlastné šablóny [7].

Rozmery plôšky pre ručné spájkovanie sú $1,55 \times 0,6$ mm, pričom pre LM358ADR je ich potreba 8 kusov v dvoch radoch. Stredy týchto plôch sú od seba vzdialené 1,27 mm a rady navzájom 5,4 mm. Plocha potrebná na napájkovanie operačného zosilňovača zaberá priestor s rozmermi $6,95 \times 4,42$ mm a plocha, ktorú zaberá celá súčiastka je $6,2 \times 5$ mm. Súčiastka je symetrická aj horizontálne aj vertikálne, viď Obr. 2.21. Component Editor umožňuje užívateľovi spravovať knižnice komponentov a vytvárať jedno alebo viaczložkové komponenty výberom šablóny a jej rozmerov, definovaním vizuálnych a elektrických parametrov pinov, nastavením modelu Spice a pripojením vzoru s 3D modelom na finalizáciu tvorby komponentov. Nástroje ako BSDL import, hromadné pomenovanie pinov a pin manager poskytujú priateľné rozhranie a maximálne možnosti pre správu pinov. Import knižníc z rôznych formátov EDA poskytuje možnosť používať existujúce zdieľané alebo „predchádzajúce“ softvérové komponenty namiesto toho, aby ich robil od nuly [7].

V module Component Editor sa nakreslila fyzickú časť operačného zosilňovača (Obr. 2.21). Tu sa však nedbalo na veľkosť, ale prispôsobili sa rozmery tak, aby sa dobre a prehľadne pracovalo pri zapájaní celej elektrickej schémy. Výstupy a vstupy opearačného zosilňovača boli pomenované podľa ich funkcie a vnútorného obvodu LM358ADR. (viď. Obr. 2.21). Posledným krokom pri navrhovaní nového komponentu je spojenie vzoru a fyzickej časti.

Ďalšou súčiastkou, ktorá sa musela vytvoriť bol servo motor. Schematickú značku som zvolila ako pre motory (Obr. 2.7). Servo má tri výstupy a to napájanie (v tomto prípade je to 5V), uzemnenie a výstup pre PWM signál. Vo vzorovom editore bolo treba teda vytvoriť len 3 diery na pájkovanie v štandardnej vzdialosti od seba 2,54 mm.

Podobné riešenie schematickej značky a vzoru bolo aj pre snímač ohybnosti. Flexi snímač má dva kolíky: jeden pre vstup 3,3V a druhý výstup. Pri tejto súčiastke sú vstup s výstupom zameniteľný, pretože snímač ohybnosti funguje podobne ako rezistor, ktorého odpor sa mení s ohybom senzora. Tak isto boli vytvorili len dve diery na napájkovanie v štandardnej vzdialosti. Snímač funguje na princípe deličky napäťia podľa čoho sme aj



Obr. 2.21: Schematický symbol operačného zosilňovača.

použili schematickú značku.

Na vyhľadanie napájania zo servo motora sa použil SMD kondenzátor veľkosti 0805. Výrobca odporúča určité rozmery na pájkovacie plôšky, ale pre pájkovanie strojom. Keďže všetky komponenty budú na šíte napájkované ručne, museli sa v dovolených medziach plochy zväčsiť. Šírka plôšky na pájkovanie bola zvolená ako štvorec so stranou 1,54 mm a vzdialenosť medzi nimi bola 0,66 mm. Schematická značka sa použila štandardná pre polarizovaný kondenzátor.

Tvorba schémy – „schematic capture“ je pokročilý nástroj pre návrh obvodov s podporou viacvrstvových a viacúrovňových hierarchických schém. Tento modul DipTrace prináša množstvo funkcií pre vizuálne a logické pripojenie pinov. Riadenie medzi modulmi zabezpečuje, že hlavné okruhy môžu byť ľahko konvertované na PCB, späťne anotované alebo importované / exportované z iných formátov EDA a CAD. Overovanie a export Spice pre simuláciu umožňujú úplnú analýzu projektu [6].

Ako prvý návrh bol vytvárený elektrický obvod pre Arduino Uno, ktoré má dve napájania 3,3 V a 5 V. Z vlastnej predpripravenej knižnice komponentov pre projekt AutomationShield [12] sa vybralo Arduino Uno. Vytvorili sa dve skupiny komponentov, ktoré patria spolu. Pre lepšiu prehľadnosť boli umiestnené na rôzne strany. V ľavo od Arduina sa nachádzal servo motor – M1. Napájanie servo motora sa pripojilo na pin 5V, uzemnenie do GND a signál do digitálneho pinu D9, ktorý dokáže prijímať PWM signál. K tomuto obvodu som ešte priradila kondenzátor C1 a diódu D1. Kondenzátor slúži na vyhľadzovanie signálu zo serva a dióda ako poistka pred spätnými prúdmi, ktoré by mohli znehodnotiť Arduino. Obidva tieto komponenty sú polarizované, to znamená, že sa musí dávať pozor na správne zapojenie. Napojili sa teda na 5 V a z druhej strany na zem.

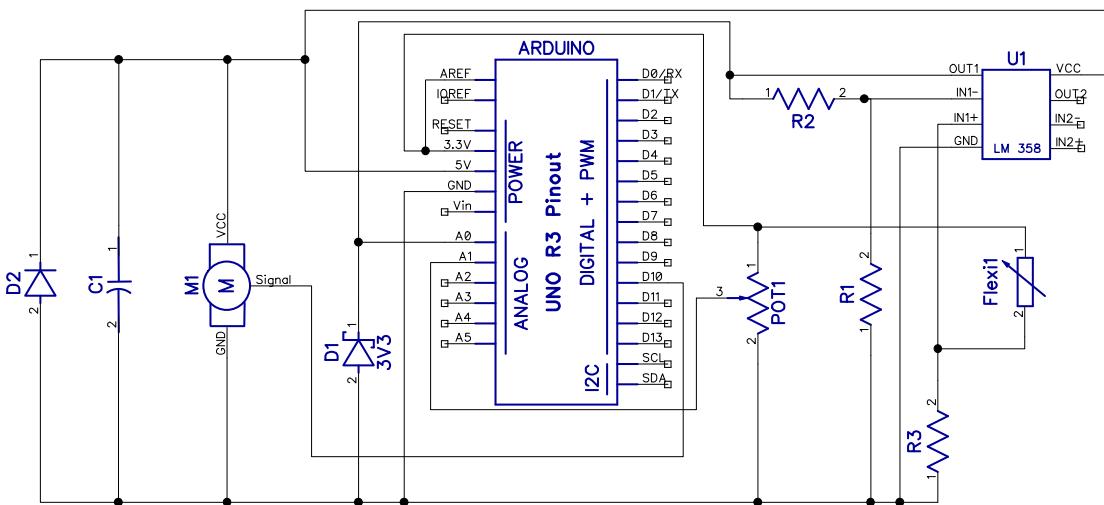
Druhá skupina komponentov obsahoval flexi snímač – U1, ktorý bol napojený na 3,3 V. Výstup zo snímača ohybnosti išiel do operačného zosilňovača a to konkrétnie do IN+. Tak isto bol tento výstup napojený do 10 $k\Omega$ rezistora – R3 a ten bol zapojený do GND, tým sa rozdelilo napätie na 1,65 a 1,65 V pri nominálnom odpore snímača. Operačný zosilňovač sa zapojil do neinvertujúceho zapojenia, čo znamená, že ak do IN+ išiel signál z flexi snímača, tak IN- bol pripojený na rezistor – R1, ktorý bol uzemnený. Z pinu číslo 1 na operačnom zosilňovači s označením OUT1 vychádzal zosilnený signál do analógového pinu A0 na Arduine. Zosilnenie bolo priamoúmerné pomeru rezistora 2 – R2 a rezistora 1 – R1, a to podľa Rov. 2.1. Rezistor R2 spája výstup OUT1 a IN-. Na operačnom zosilňovači boli ešte pin na napájanie a pin na GND, ktoré tiež museli byť zapojené. LM358ADR obsahuje ešte jeden zosilovač, ale ten neboli v našom obvode využitý.

Posledný komponent, ktorý bol umiestnený do schémy bol potenciometer. Ani ten neboli v knižnici čo obsahoval DipTrace. Znova bola použitá vlastná knižnica komponentov

AutomationShield [12]. Potenciometer – POT1 sme pripojili na 5V, GND a analógový pin A1.

V dnešnej dobre je trend znižovať pracovné napätie procesorov, a preto moderné napájanie MCU majú logiku 3,3 V (CMOS) a nie 5 V (TTL). Dôvodom je lepšia technológia výroby čipov a úspora energie. TugShield som sa snažila vytvoriť tak aby kompatibilný so všetkými doskami, ktoré majú rovnaké rozhranie pinov. Celé je to zastrešené do triedy R3, kde patria okrem Arduina Una aj Arduino Zero, Arduino Metro a iné.

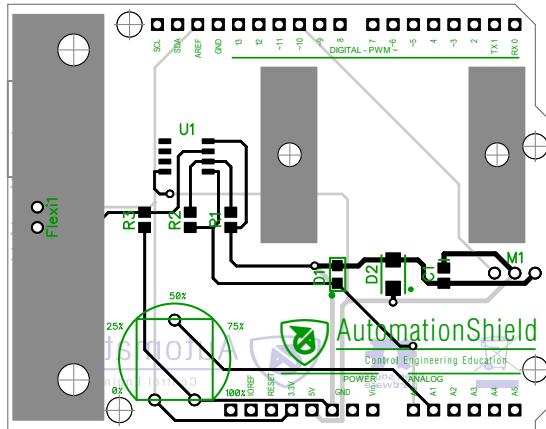
Pôvodný plán bol napojiť operačný zosilňovač miesto 5 V na 3,3 V, ale LM358 by nepracoval správne. Do obvodu sme teda zapojili aj zenerovu diódu, ktorá nám nedovoľovala väčší prúd na pine A0 a operačný zosilovač bol naďalej napájaný piatimi voltami (Obr. 2.22).



Obr. 2.22: Schéma zapojenia.

Po vyhotovení schémy sa pomocou modulu PCB Layout vygenerovala do PCB vzhľadu. PCB Layout je vysoko-úrovňový inžiniersky nástroj pre plošných spojov, ktorý obsahuje inteligentné manuálne smerovanie vysokorychlostných a diferenciálnych signálov, tvarovo založený autorouter, pokročilé overovanie a široké možnosti importu a exportu. Požiadavky na návrh sú definované triedami, pravidlami triedy a podrobnnými nastaveniami podľa typov objektov pre každú triedu alebo vrstvu. DipTrace funkcie návrhu procesu s real-time DRC (Design Rule Check, v preklade kontrola návrhu), ktorý hlási chyby pred skutočným výrobením. Doska sa dá zobraziť v 3D a exportovať pre mechanické CAD modelovanie [5].

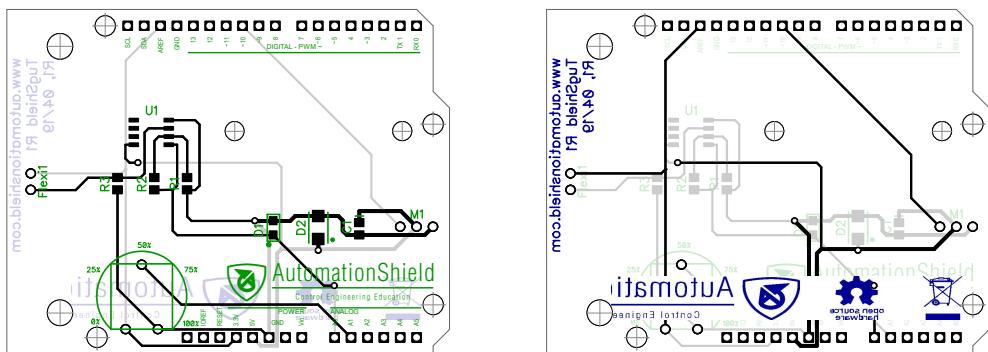
V PCB Layoute sa automaticky vytvoril PCB návrh podľa schémy zapojenia na Obr. 2.22. Komponenty neboli zobrazené s ich fyzickým rozhraním, ale boli znázornené plochy kde sa napájkujú reálne súčiastky. Po prvom vygenerovaní boli zapojenia medzi plochami priamočiare. Úlohou bolo usporiadať súčiastky tak, aby z toho vznikol požadovaný shield. Všetky komponenty sa museli nachádať v priestore prototypovej dosky, ktorú tvoril tvar Arduina Una. Rozmiestnili sa tak, aby sa všetky zmestili a navzájom sa neprekryvali alebo si nezavadzali, ale tak isto sa muselo dbať aj na to, aby komponenty, ktoré patria k sebe, neboli od seba ďaleko. Takže vznikli dva súbory plôch na spájkovanie (Obr. 2.23).



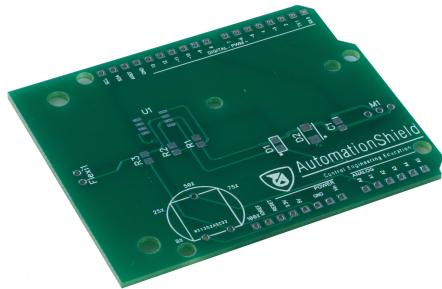
Obr. 2.23: Rozmiestnenie komponentov na PCB dosku.

Na upevnenie serva bol použitý mostík, ktorý sa naspodu uchytí dvomi skrutkami. Preto sa do návrhu PCB umiestnil aj naznačenie miesta kde sa bude nosník na servo dotýkať základovej dosky a rozmiestnili sa aj diery pre skrutky, ktoré budú automaticky predvŕtané od výrobcu. Rovnako som postupovala aj pri zvernom spoji, ktorý tvorí L - kový tvar upevnený dvoma skutkami o dosku a ďalšia doska, ktorá zvieria flezi snímač s nosníkom.

Po vygenerovaní automatických trás sa zobrazili všetky priamočiare cesty zapojenia v reálnom tvaru. Obvod sa precízne prekontroloval. Snaha bola, aby zalomenia spojov nevytvárali pravý uhol, ale aby sa zalomili v dvoch bodoch, ktoré vytvorili skosený roh. Posledným krokom bolo na dosku umiestniť vhodnú a potrebnú potlač. Názov projektu – AutomationShield – bol na prednej časti a na spodnej symbol, že hardvér sa nemôže vyhadzovať do bežného odpadkového koša, link pre viacej informácií, názov shieldu - TugShield, číslo verzie a dátum (Obr. 2.24). Po konečnom skontrolovaní sa odstránilo zobrazenie najvyššej vrstvy PCB dosky, kde boli naznačené umiestnenia nosníkov na doske (viď. Obr. 2.24) a doska sa odoslala v štandardnom formáte CAD "Gerber" do výroby. Hotový výrobok PCB dosky je na Obr. 2.25.



Obr. 2.24: Predná a zadná strana finálnej PCB dosky.



Obr. 2.25: PCB doska TugShieldu.

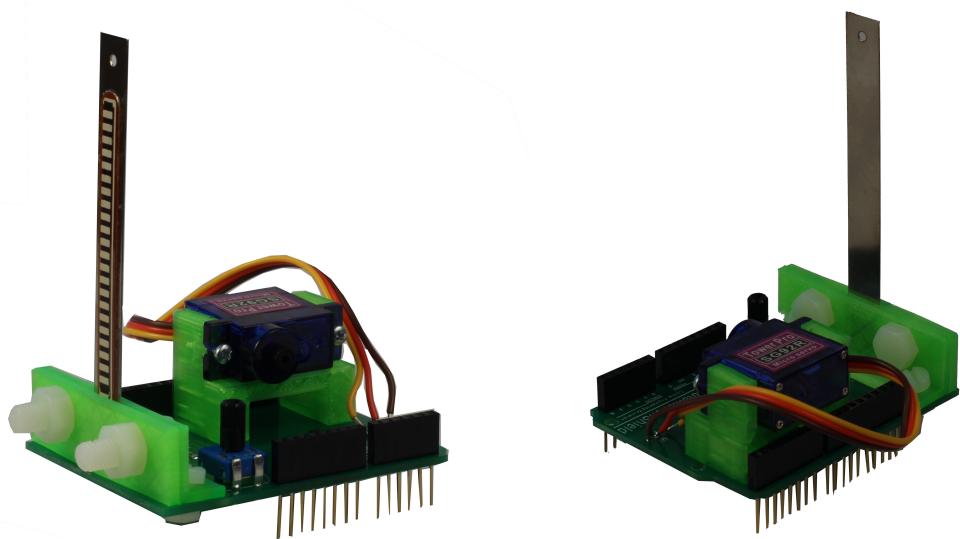
2.4 Konečný model TugShieldu

Vo finálnom výrobku sa nepoužila popísaná prototypová doska v Kap. 2.1.1. Z praktických a estetických dôvodov bola použitá navrhnutá, ale strojom vyrobená PCB dosku z Kap. 2.3. Obvod nebol napájkovaný ručne.

Hlavným cieľom návrhu pri vytváraní dosiek bolo čo najlacnejšie, najjednoduchšie a najuniverzálnejšie riešenie. Snaha bola vylúčiť drahé mechanické alebo elektrické komponenty, aby sa štít mohol používať na didaktické účely a bol dostupný aj pre organizácie s nízkym rozpočtom. Najdrahším komponentom bol jednoznačne flexi snímač, ktorý stál priemerne 10 krát viacej ako ostatné súčiastky (viď. Tab. 2.3).

Tabuľka 2.3: Cena a informácie o použitých komponentoch

Komponent	Výrobca	Výrobné číslo	Cena (EUR)
Operačný zosilňovač	Texas Instruments	LM358ADR	0,31
Zenerova dióda	SEMTECH	BZX 55/C 3V3	0,03
Usmerňovač	Vishay Semiconductors	RGF1D-E3/67A	0,41
Rezistor	ROHM Semiconductor	SFR10EZPJ103	0,09
Flexi snímač	SparkFun	SEN-10264	7,94
Microservo	Adafruit	169	5,2
Kondenzátor	AVX	F980J107MSA	0,58
Potenciometer	ACP	CA6H 2K5	0,12
PCB doska	PCBWay	-	0,5
-	-	Celková cena	15,68



Obr. 2.26: Finálny produkt.

3 Tvorba softvéru

Vytvorenie softvérovej časti TugShieldu bolo rovnako dôležité ako hardvérovej. Mojou úlohou bolo pomocou programátorského rozhrania aplikácie (API – Application Program Interface) vytvoriť ovládací softvér a inštruktážne príklady. Táto kapitola je zameraná na ovládací softvér.

Programátorské rozhranie aplikácie (API) je súbor metód, protokolov a nástrojov na vytváranie softvérových aplikácií [32]. API udáva, ako majú softvérové komponenty na vzájom pôsobiť. Nepoužíva sa len na vytvorenie softvérového rozhrania, ale tak isto sa využíva aj pri programovaní komponentov grafického rozhrania. Uľahčuje vývoj programu tým, že poskytuje základné “stavebné bloky”, ktoré programátor spojí dohromady a vytvorí tým logický program [32].

Vytvorenie softvérového rozhrania pre TugShield má samo o sebe náučnú hodnotu. Študenti môžu získať skúsenosti z návrhu a konštrukcie programu a obohatí sa im programátorská logika. Na súčasnom trhu práce poskytuje ovládanie programátorskej logiky veľkú výhodu absolventom vysokých škôl, pretože je to veľmi žiadana schopnosť.

Bohužiaľ, pri výuke je mälokedy čas venovať sa vytváraniu rozhrania. Na cvičenia, kde sa študenti učia základy programovania, by bola táto časť návrhu experimentálneho zariadenia veľmi obohacujúca, ale na cvičeniach ako napríklad teória riadenia, vyučujúci preferujú už hotové softvérové rozhranie.

Z tohto dôvodu som vytvorila rozhranie pre programovanie aplikácií v jazyku C/C++ pre Arduino IDE, aby bol Tugshield hned použiteľný vo vyučovacom procese.

3.1 Trieda TugShield

Trieda `TugShield` obsahuje ovládače hardvéru. API so všetkými metódami sú zahrnuté v súbore `TugShield.h`. Spomenutá trieda `TugShieldClass` je štandardne konštruktovaná ako objekt `TugShield`.

Na začiatku súboru `TugShield` je preventívne opatrenie, ktoré zabraňuje, aby sa knižnica vo prostredí Arduino IDE zavolala viackrát. Spôsobilo by to viacnásobnú deklaráciu.

```
#ifndef TUGSHIELD_H_
#define TUGSHIELD_H_
```

Súbor `TugShield.h` v sebe zahŕňa ešte hlavičky, ktoré sú taktiež použité pri tvorení API pre `TugShield`.

```
#include "Arduino.h"
#include "AutomationShield.h"
#include "Servo.h"
```

```
#include "Sampling.h"
```

V ďalšej časti programu sa definujú piny, ktoré sa používajú na hardvéri a predefinované konštanty. Pre analógový výstup z flexi snímača (**TUG_YPIN**) je použitý analógový pin 0. Servo motor sa ovláda cez digitálny pin 10 (**TUG_UPIN**). Dalej sú tu definované konštatnty ako **K_MIN**, **K_MAX**, ktoré používajú podmienky v kalibrácií. **DEFAULT_MIN** a **DEFAULT_MAX** sú prednastavené hodnoty, v prípade, že by kalibrácia neprebehla.

```
#define TUG_YPIN 0
#define TUG_UPIN 10
#define K_MIN 1023
#define K_MAX 0
#define DELAY_VALUE 500
#define SERVO_MIN 0
#define SERVO_MAX 180
#define DEFAULT_MIN 500.0
#define DEFAULT_MAX 1000.0
```

Vytvorila sa trieda s názvom **TugShieldClass**, ktorá v sebe obsahuje 4 metódy dostupné pre používateľa. Metóda **Begin** má za úlohu inicializovať hardvér TugShiledu a následne na to ho kalibruje metóda **Calibration**. Metódy **ActuatorWrite** a **Sensor Read** zabezpečujú získavanie vstupno-výstupných hodnôt. V triede sú zadefinované aj 4 globálne premenné, ale s tými používateľ nemôže pracovať.

```
class TugShieldClass
{
public:
    void begin();
    void calibration();
    void actuatorWrite(int servo_angle);
    float sensorRead();
private:
    bool _wasCalibrated;
    int _sensorRead ;
    int k_maximal;
    int k_minimal;
};
```

Ako posledný je vytvorený objekt triedy TugShiled s názvom **TugShield**.

```
extern TugShieldClass TugShield;
```

3.2 Metóda Begin

Metóda **Begin** už podľa názvu begin - začať, má za úlohu na začiatku programu inicializovať hardvér.

Pomocou príkazu **servo.attach(TUG_UPIN)** sa nastaví pin, ktorý je zapojený ako digitálny vstup do servo motora (v našom prípade je to digitálny pin číslo 10 na Arduino Uno). Cez tento pin sa nastavuje poloha serva, čo sa hned využíva v druhom

riadku kódu `servo.write(SERVO_MAX)`. Na začiatku sa nastaví servo do polohy 0°. Poloha 0° pre TugShield je definovaná ako poloha, ktorá vytvára minimálny ohyb nosníka. V príkaze sa zadáva ako hodnota 180°, pretože servo je horizontálne naopak uložené.

Ako posledné sa v metóde Begin nachádza príkaz na nastavenie externej analógovej referencie, ktorú sme vytvorili pripojením napájania 3,3 V s pinom AREF. Tento krok musí byť ale sprevádzaný aj so softvérovým príkazom `analogReference(EXTERNAL)`.

Užívateľ použije metódu Begin príkazom `TugShield.begin()` v prostredí Arduino IDE.

```
void TugShieldClass::begin() {
    servo.attach(TUG_UPIN);
    servo.write(SERVO_MAX);
    analogReference(EXTERNAL);
}
```

3.3 Metóda Calibration

Po inicializovaní hardvéru nasleduje metóda na kalibrovanie. Cieľom metódy je zistiť najväčšie hodnoty výstupu flexi snímača na analógovom prevodníku pri polohe servo motora 0° a najnižšie pri polohe 180°. Ako bolo spomenuté v Kap. 3.2, poloha 0° pre TugShield je poloha, v ktorej je flexi snímač čo najmenej ohnutý. Snímač ohybnosti tu vykazuje najvyššie hodnoty odporu, čo je spôsobené zapojením (viď. Kap. 2.3). Do príkazu `servo.write()` sa musí zapísat hodnota 180°, ktorá je uložená pod konštantou `SERVO_MAX`.

Na začiatku sa vypíše informácia pre používateľa "Prebieha kalibracia". Po nastavení servo motora sa zistuje najvyššia hodnota pomocou cyklu. For cyklus prebehne 11x , v prvom cykle sa nameraná hodnota z flexi snímača porovnáva s vopred nastavenou hodnotou `K_MAX`. Hodnota konštatny `K_MAX` bola nastavená tak, aby podmienka bola vždy splnená. Potom nasleduje opakovanie cyklu zvyšných 10 x, kedy sa porovnávajú navzájom namerané hodnoty z flexi snímača. Nameraná hodnota sa vždy porovná s predchádzajúcou a ak je väčšia, zapíše sa do premennej `k_maximal`. Touto metódou sa zistí najväčšia hodnota z flexi senzora.

Tá istá logika sa použije aj pri kalibrovaní polohy servo motora 180°. Do príkazu `servo.write()` sa zapíše konštantu `SERVO_MIN`, ktorá je rovná 0°. Potom sa hľadá najmenšia hodnota odporu flexi snímača. Pre TugShiled v tomto prevedení sa hodnoty premenej `k_maximal` pohybujú v rozmedzí od 980 - 1010 A/D úrovni a pre `k_minimal` je to interval hodnôt od 440 - 470 A/D úrovni.

Na konci metódy sa vypíše text "Kalibracia bola uspesne dokoncena" a premennej `_wasCalibrated`, typu `bool` sa priradí 1. Premenná sa používa na zistenie, či kalibrácia prebehla úspešne alebo je nutné použiť prenastavené hodnoty. Kalibrácia sa vykoná zadaním príkazu `TugShield.calibration()` do prostredia Arduino IDE.

```
void TugShieldClass::calibration() {
    Serial.println("Prebieha kalibracia");
    servo.write(SERVO_MAX);
    delay(DELAY_VALUE);
```

```

        for (int ii=0; ii <= 10; ii++) {
            _sensorRead=analogRead(TUG_YPIN);
            if (ii == 00) {
                if(_sensorRead>K_MAX) {
                    k_maximal=_sensorRead;
                    delay(DELAY_VALUE);
                }
            } else {
                if(_sensorRead>k_maximal) {
                    k_maximal=_sensorRead;
                    delay(DELAY_VALUE);
                }
            }
        }
        servo.write(SERVO_MIN);
        delay(DELAY_VALUE);
        for (int ii=0; ii <= 10; ii++) {
            _sensorRead=analogRead(TUG_YPIN);
            if (ii == 00) {
                if(_sensorRead<K_MIN) {
                    k_minimal=_sensorRead;
                    delay(DELAY_VALUE);
                }
            } else {
                if(_sensorRead<k_minimal)
                    k_minimal=_sensorRead;
                delay(DELAY_VALUE);
            }
        }
        _wasCalibrated = true;
        Serial.println("Kalibracia bola uspesne dokoncena");
    }
}

```

3.4 Metóda ActuatorWrite

V triede TugShield je zahrnutá aj metóda ActuatorWrite. Do tejto metódy používateľ musí zadať hodnotu polohy servo motora, teda uhol, do akého sa má rameno serva nastaviť. V druhom riadku metódy sa zadaná hodnota prepočíta na relatívnu hodnotu (viď. Kap. 3.2) a tá sa následne zapíše do príkazu `servo.write()`. Premenná `servo_angle`, ktorú používa úžívateľ aj `servo_rightangle`, sú celočíselné premenné (typ `int`). Metóda ActuatorWrite sa použije príkazom `TugShield.calibration(int servo_angle)`.

```

void TugShieldClass::actuatorWrite(int servo_angle) {
    int servo_rightangle = 180 - servo_angle;
    servo.write(servo_rightangle);
}

```

```
}
```

3.5 Metóda SensorRead

Poslednou nevyhnutnou metódou na pracovanie s TugShieldom je SensorRead. SensorRead je ako jediná z metód, ktorá má výstupný charakter, čo znamená, že v poslednom riadku sa nachádza príkaz `return`. Po zadaní príkazu `TugShield.sensorRead()` a prebehnutí metódy, Arduino IDE vráti premennú `_sensor_rightValue` typu `float`.

Na začiatku metódy sa do premennej `_sensorRead` typu `int` načíta hodnota úrovne A/D z flexi snímača. V druhom riadku sa definovala premenná `_sensorValue`, ktorá je následne použitá v podmeinke. Podmienka kontrolouje, či kalibrácia prebehla alebo nie. Ak áno, tak na premapovanie použije premenné `k_maximal` a `k_minimal`. Pokiaľ nie, budú použité prednastavené hodnoty uložené v konštantách `DEFAULT_MIN` a `DEFAULT_MAX`.

Cieľom metódy SensorRead je prečítať hodnotu z flexi snímača a previesť ju na percentá. V rozhraní Arduino IDE sa zadá ako `TugShield.sensorRead()`.

```
float TugShieldClass::sensorRead() {
    _sensorRead = analogRead(TUG_YPIN);
    float _sensorValue;
    if(_wasCalibrated == true) {
        _sensorValue = AutomationShield.mapFloat(
            ((float)_sensorRead,(float)k_minimal,
            (float)k_maximal, 0.00, 100.00);
    }
    else {
        _sensorValue = AutomationShield.mapFloat(
            (float)_sensorRead, DEFAULT_MIN,
            DEFAULT_MAX, 0.00, 100.00);
    }
    float _sensor_rightValue = 100 - _sensorValue;
    return _sensor_rightValue;
}
```

4 Demonštračné príklady

Demonštratívne príklady slúžia jednako na otestovanie funkčnosti hardvéru a zároveň na získanie vstupno-výstupných dát, ktoré môžu byť ďalej použité na prenosové funkcie. V nasledujúcej kapitole sa uvádzajú tri príklady na objasnenie fungovania TugShieldu.

4.1 Príklad identifikácie systému TugShield

Príklad identifikácie systému TugShield je príklad, ktorý slúži na získanie vstupno-výstupných dát pomocou vytvorenia preddefinovaného profilu vstupov v otvorenej slučke. Využíva podsystém vzorkovania z knižnice AutomationShield za účelom zabezpečenia vzorkovania na základe prerušení v reálnom čase.

V prvom rade sa musí zahrnúť hlavička TugShield, ktorá je obsiahnutá v súbore **TugShield.h**. Následne sa definujú potrebné premenné, ktoré sa budú v príklade využívať. **Ts** je premenná typu **unsigned long**, ktorá udáva periódu vzorkovania. Zadáva sa v milisekundách a po testovaní sa zistilo, že vyhovujúca hodnota pre túto premennú je 10. Premenná **k** typu **unsigned long** počíta počet vzoriek a na inicializáciu potvrdenia kroku sa definovala premenná typu **bool - enable**.

Výstup z flexi snímača sa ukladá do premennej **y** typu **float**. Na zapísanie uhu ramena servo motora sa vytvorila celočíselná premenná **u**. Trajektória pozícii ramena servo motora je uložená v celočíselnom poli **U[]**. Ósmy riadok príkladu udáva počet vzoriek pri jednej pozícii servo motora a deviaty definuje premennú na počítanie segmentov experimentu.

Príklad sa začne inicializovaním sériovej komunikácie, inicializovaním a kalibrovaním hardvéru TugShield. Nastaví sa inicializácia vzorkovania a zadefiniuje sa funkcia na prerušenie vzorkovania.

Vytvorí sa slučka, ktorá spúšťa funkciu **step()** vtedy, keď premenná **enable** je rovná hodnote **true**. O tom rozhoduje funkcia **stepEnable()**, ktorá sleduje, či prešiel dostatočný čas na ďalšiu vzorku.

Kroková funkcia postupne načítava do premennej **u** hodnoty z poľa **U[]**. Premenná **u** sa zapíše do príkazu **TugShield.actuatorWrite(u)**, a tým sa rameno serva nastaví do želanej pozície. Príkazom **TugShield.sensorRead()** sa odčíta výstupná hodnota z flexi snímača a priradí sa premennej **y**.

```
#include "TugShield.h"
unsigned long Ts = 10;
unsigned long k = 0;
bool enable=false;
```

```

float y = 0.0;
int u = 0;
int U[]={0,90,45,100,180,0};
int T = 100;
int i = 0;

void setup() {
Serial.begin(9600);
TugShield.begin();
TugShield.calibration();
Sampling.period(Ts * 1000);
Sampling.interrupt(stepEnable);}

void loop() {
if (enable) {
step();
enable=false; }
void stepEnable(){
enable=true; }

void step(){
if (k % (T*i) == 0){
u = U[i];
i++;}

TugShield.actuatorWrite(u);
y = TugShield.sensorRead();
Serial.print(y);
Serial.print(", ");
Serial.println(u);
k++; }

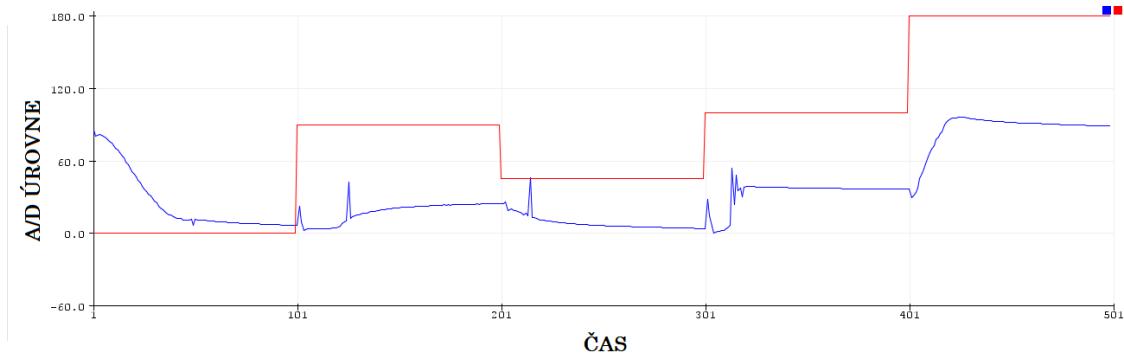
```

Na konci programu sa aj vstupná premenná `u` aj výstupná premenná `y` zapíšu do sériového zapisovača. Výsledkom príkladu je graf, ktorý zobrazuje meniace sa vstupno-výstupné dátá (Obr. 4.1).

4.2 Príklad s aplikovaním PID regulátora.

Druhý demonštračný príklad má za úlohu riadiť spätnú väzbu ohybu nosníka pomocou PID regulátora. Inicializujú sa vzorkovacie a PID riadiace podsystémy z knižnice AutomationShield a spustí sa vopred určená trajektória žiadanej deformácie.

Príklad s aplikovaním PID regulátora je programovo podobný s príkladom v Kap. 4.1. Na začiatku programu sa zahrnie knižnica TugShield. Definujú sa premenné `Ts`, `k`, `enable`, `T`, `i`, `y` a `u`, ktoré majú rovnakú funkciu ako aj v príklade na identifikáciu systému. Premenná `r` je žiadaná hodnota regulácie. Vytvorí sa celočíselné pole `R`, z ktorého sa budú načítavať hodnoty pozície ramena servo motora.



Obr. 4.1: Výstup z príkladu identifikácie systému.

Na PID regulátor potrebujeme tri regulačné konštanty (KP, TI a TD), ktorým sa priradia hodnoty podľa testovania. Tie sa na nastavia pomocou príkazov `PIDAbs.setKp(KP)`, `PIDAbs.setTi(TI)` a `PIDAbs.setTd(TD)`. Cieľom je, aby výstupná hodnota, čo najlepšie sledovala žiadanú hodnotu.

Nasleduje inicializovanie sériovej komunikácie, inicializovanie hardvéru TugShield a jeho kalibrácia. Nastaví sa períoda vzorkovania a vytvorí sa funkcia na prerušenie.

Podmienka v riadku 29 spúšťa funkciu `step()` podľa premennej `enable`, ktorá má hodnotu true iba vtedy, keď prejde dostatok času od poslednej vzorky. To zabezpečuje funkcia `stepEnable()`, ktorá počíta počet vzoriek.

Ak je krok schválený, z pola `R[i]` sa načíta hodnota do premennej `r`. Urobí sa rozdiel premenných `r` a `y`, ktorý sa použije v príkaze `PIDAbs.compute()`. Nakoniec sa nová zistená hodnota `u` zapíše do servo motora.

```
#include "TugShield.h"

unsigned long Ts = 10;
unsigned long k = 0;
bool enable=false;

int r = 0;
int R[]={0,90,45,100,180,0};
int T = 100;
int i = i;
int y = 0;
int u ;

#define KP 0.5
#define TI 0.1
#define TD 0.0

void setup() {
    Serial.begin(9600);
    TugShield.begin();
    TugShield.calibration();
```

```

Sampling.period(Ts * 1000);
Sampling.interrupt(stepEnable);
PIDAbs.setKp(KP);
PIDAbs.setTi(TI);
PIDAbs.setTd(TD); }

void loop() {
    if (enable) {
        step();
        enable=false;}}
```

```

void stepEnable(){
    enable=true; }

void step(){
    if (k % (T*i) == 0){
        r = R[i];
        i++;}

u = PIDAbs.compute(r-y,0,180,0,100);
TugShield.actuatorWrite((int)u);
y = TugShield.sensorRead();

Serial.print(r);
Serial.print(",\"");
Serial.print(y);
Serial.print(",\"");
Serial.println(u);
k++;
}

```

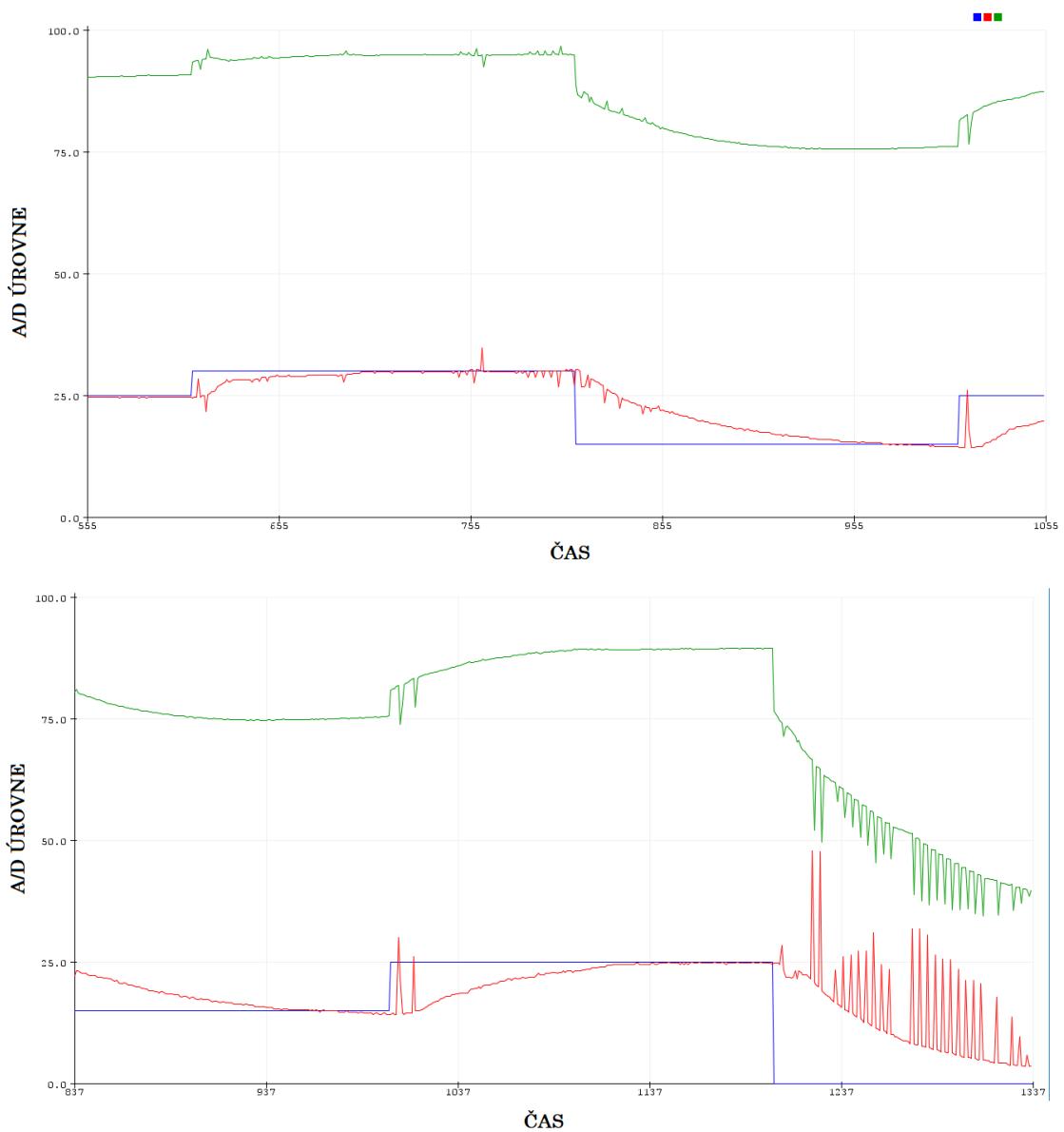
Výsledkom príkladu s použitím PID regulátora je graf, na ktorom je jasne vidieť zregulovanú hodnotu 4.2. Ako zosilnenie sme použili hodnotu 0,5, integračnú konštantu 0,1 a derivačná bola nastavená ako nulová. To z dôvodu, že nepresnosť serva spôsobovalo náhle skokové zmeny a derivácia miesta skokovej zmeny nadobúda hodnoty nekonečna. Pri meraní sa zistilo, že ide o nelineárny systém, dôkazom čoho je Obr. 4.3. Pozície servo motora boli v tomto teste väčšie ako 35° a regulácia už nebola možná.

4.3 Príklad Selftest

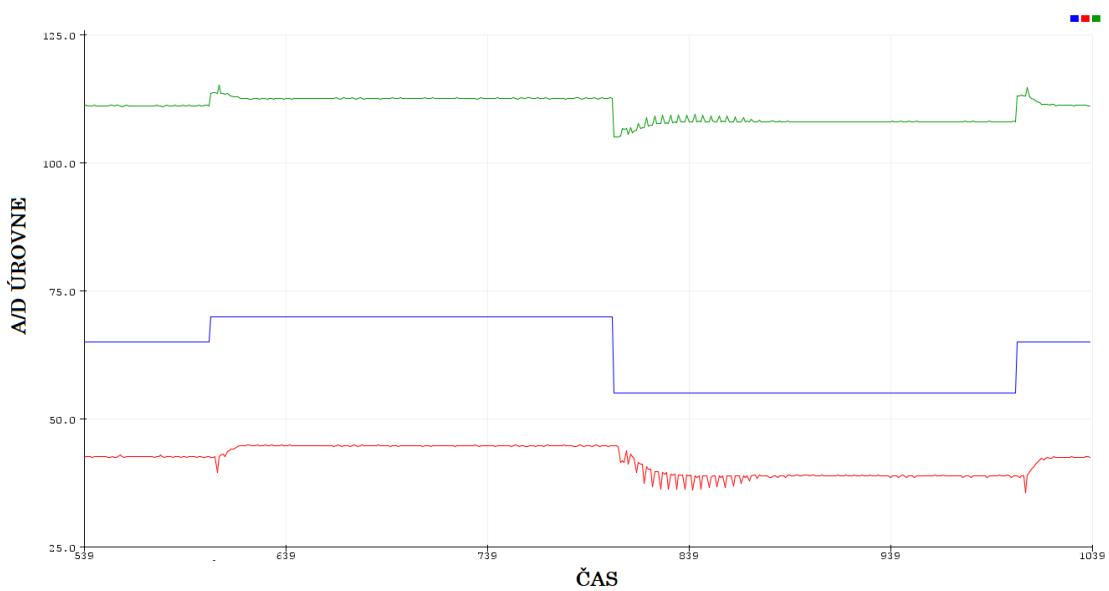
Príklad “Selftest” slúži na pretestovanie hardvéru, správnosti kalibrácie a zistenie číselných hodnôt merania.

Ako prvé sa zahrnie hlavička TugShield, ktorá je zahrnutá v súbore TugShield.h. Definuje sa premenná y typu `float`, ktorá bude načítavať výstup z flexi snímača.

Nutné je inicializovať sériovú komunikáciu príkazom `Serial.begin()`, a tak isto inicializovať a nakalibrovať hardvér TugShield príkazmi `TugShield.begin()` a `TugShield`



Obr. 4.2: Výstup z príkladu PID.



Obr. 4.3: Výstup z príkladu PID pri pozícii serva väčších ako 35° .

`.calibration()`.

Na monitor sériového portu sa vypíše správa pre používateľa, ktorá poloha sa ide testovať. Ako prvú testuje program polohu serva 0° . Príkazom `TugShield.actuatorWrite(0)` sa nastaví pozícia a následne sa pomocou `TugShield.sensorRead()` odčíta hodnota z flexi snímača. Priradí sa do premennej `y`. V podmienke v riadku 15 sa zistí, či nameraná hodnota splňa kritéria alebo nie.

Rovnaký postup sa použije aj pre pozíciu ramena servo motora 180° . Pokiaľ hodnoty splňajú kritéria, na monitore sériového portu sa vypíše “V poriadku.”. Pokiaľ nie, tak “Nevyhovuje.”. Pre informáciu sa zobrazia aj namerané hodnoty. Výstup z príkladu Self-Test je zobrazený na Obr. 4.4.

```
#include "TugShield.h"
float y = 0.0;

void setup()
{
Serial.begin(9600);
TugShield.begin();
TugShield.calibration();
delay(100);

Serial.println("Testovanie flexi snimaca... poloha serva: 0");
TugShield.actuatorWrite(0);
delay(500);
y = TugShield.sensorRead();
if (y >= 0.0 && y <= 15.0) {
Serial.println("V poriadku.");
}
else {
Serial.println("Nevyhovuje.");
}
}
```

```

Serial.println(y);}
else {
Serial.println("Nevyhovuje.");
Serial.println(y); }
delay(1000);

Serial.println("Testovanie flexi snímaca... poloha serva: " +
180);
TugShield.actuatorWrite(180);
delay(500);
y = TugShield.sensorRead();
if (y >= 85.0 && y <= 100.0) {
Serial.println("V poriadku.");
Serial.println(y);}
else {
Serial.println("Nevyhovuje.");
Serial.println(y);}
}

```

Prebieha kalibrácia
Kalibrácia bola úspešne dokončená
Testovanie flexi snímača... poloha serva: 0
V poriadku.
10.61
Testovanie flexi snímača... poloha serva: 180
V poriadku.
96.93

Obr. 4.4: Výstup z príkladu Self-Test.

Príklady v Kap. 4.1 až 4.3 plne otestovali funkčnosť hardvéru. Získali sa vstupno-výstupné údaje v podobe grafov, ktoré sa ďalej môžu použiť na prenosové funkcie.

5 Záver

Cieľom záverečnej práce bolo navrhnúť, skonštruovať a spojazdniť experimentálne zariadenie na riadenie statickej deformácie votknutého nosníka. Tugshield sa skladal z dvoch základných komponentov a to nosník, na ktorom bol upevnený flexi snímač a servo motor. Vytvorenie rozhrania API poskytovalo metódy na ovládanie hardvéru, čím bolo možné získať vstupno-výstupné hodnoty riadenia deformácie.

Prvá kapitola sa zaoberala situáciou vo svete ohľadom implementácií Arduina do vzdelávacieho procesu. V rôznych krajinách boli vytvorené originálne projekty so zmysluplnými nápadmi. Pri prieskume sa zistilo, že ich veľká nevýhoda bola neschopnosť reprodukovať a použiť projekty aj na iných miestach, ako tam kde vznikli. Dôvodom bola nedostatočná dokumentácia, vo veľa prípadoch žiadna. Kapitola sa zaoberala aj situáciou na Slovensku.

V druhej kapitole sa opisuje postup pri navrhovaní fyzickej časti TugShieldu. V prvom rade sú to použité súčiastky, ich parametre a potrebné údaje. Vývoj prototypov od prvého až po finálny produkt, návrh PCB dosky a elektrického obvodu. Riešila sa aj problematika ohľadom zosilnenia operačného zosilňovača, ktorý pracoval z výstupným signálom z flexi snímača. Navrhla sa najlepšia možná kombinácia rezistorov pre neinvertujúci operačný zosilňovač. Na vytvorenie finálneho produktu sa museli vyrobiť 3 prototypy, ktoré postupne odhaľovali nedostatky a problémy, ktoré boli na finálnom TugShielde odstranené.

Návrh programového rozhrania (API) sa začalo vytvorením metód na ovládanie hardvéru. Na inicializáciu a kalibráciu hardvéru sa vytvorili metódy `Begin` a `Calibration`. Ďalšou metódou bola `ActuatorWrite`, ktorá vpisovala zadané hodnoty do servo motora a poslednou bola metóda na odčítavanie výstupných hodnôt z flexi snímača `SensorRead`. Vytvorila sa trieda s názvom `TugShield`, ktorá zastrešovala všetky metódy.

V poslednej kapitole sa prezentovali demonštračné príklady, ktorých výsledkom boli vstupno-výstupné hodnoty zobrazené na grafoch. Príklady testovali hardvér a vytvorené rozhranie (API). Pomocou PI regulátora sa zregulovali výstupné hodnoty flexi snímača a z výsledku vyplynulo, že systém je nelineárny a pri vyšších hodnotách sa nedá riadiť pomocou PID regulátora.

Na základe výsledkov z jednotlivých kapitol môžeme skonštatovať, že navrhnutie hardvéru a aj programového rozhrania sa podarili.

5.1 Návrhy na zlepšenie

Návrh zapojenia flexi snímača ako delič napäťia nie je ideálny. Od napájania je prvý zapojený flexi snímač a až potom rezistor. Spôsobuje to obrátenú logiku výstupných hodnôt z flexi snímača, a to že čím je snímač viac ohnutý, tým je hodnota úrovne A/D nižšia. V budúcnosti by si v zapojení mali vymeniť poradie.

Ďalším nedostatkom bol operačný zosilňovač LM358ADR, ktorý neboli “rail-to-rail” a preto nemohol byť napájaný z 3,3 V. Napájaný bol z 5 V, čo mohlo spôsobiť napätie na analógovom pine väčšie ako 3,3 V. V elektrickom obvode pre TugShield to znamenalo, že bolo nevyhnutné použiť aj zenerovú diódu so zenerovým napäťom 3,3 V.

Softvérový program by mohol byť v budúcnosti rozšírený aj o MATLAB a Simulink.

Čo sa týka umiestnenia servo motora na TugShield, na ďalšom prototype by sa mal prevrátiť horizontálne o 180 stupňov. Hodnoty pozície ramena by sa mohli zadávať priamo bez medzivýpočtu. Tak isto chýba označenie pinov pre servo motor, pri pájkovaní to zneprehľadňuje a spomaľuje robotu.

Plôšky určené na napájkovanie kondenzátora, boli navrhnuté príliš malé. Komponent je určený na pájkovanie strojom a nie ručne.

Zosilnenie signálu z flexi snímača bolo zosilnené pomocou neinvertujúcej konfigurácie, ale vhodnejšie by bola diferenčné zapojenie.

Samozrejme zlepšiť by sa mohol aj dizajn komponentov vytlačených na 3D tlačiarni. Jednoduchšie a krajšie prevedenie by bolo vítané.

Literatúra

- [1] Adafruit. Micro servo. Online, 2019. [cit. 5.4.2019], <https://www.adafruit.com/product/169>.
- [2] Arduino. Reading a Potentiometer (analog input). Online, 2019. [cit. 24.3.2019], <https://www.arduino.cc/en/tutorial/potentiometer>.
- [3] B. R. Barčík. *Meranie charakteristík operačných zosilňovačov pomocou LabVIEW*. PhD thesis, Žilinská univerzita v Žiline, Elektrotechnická fakulta, Žilina, Slovenská Republika, Máj 2006.
- [4] Curtreview. WINGONEER Prototype PCB for Arduino UNO R3 Shield Board DIY. Online, 2019. [cit. 27.4.2019], <https://www.curtreview.com/fr/B01FTVTJT2.html>.
- [5] DipTrace. Advanced Routing Capabilities. Online, 2019. [cit. 27.4.2019], <https://diptrace.com/diptrace-software/pcb-layout/>.
- [6] DipTrace. Multi-Level Hierarchical Schematics. Online, 2019. [cit. 27.4.2019], <https://diptrace.com/diptrace-software/schematic-capture/>.
- [7] DipTrace. Smart Component Management. Online, 2019. [cit. 27.4.2019], <https://diptrace.com/diptrace-software/library-creation/>.
- [8] W. J. Esposito, F. A. Mujica, D. G. Garcia, and G. T. A. Kovacs. The Lab-In-A-Box project: An Arduino compatible signals and electronics teaching system. In *2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, pages 301–306, Aug 2015.
- [9] Fluke. What is a diode? Online, 2019. [cit. 22.5.2019], <https://www.fluke.com/en-us/learn/best-practices/measurement-basics/electricity/what-is-a-diode>.
- [10] C. Y. Foulis and S. Papadopoulou. A Portable Low-Cost Arduino-Based Laboratory Kit for Control Education. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 435–435, Sep. 2018.
- [11] A. A. Galadima. Arduino as a learning tool. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, pages 1–4, Sep. 2014.
- [12] Gergely Takács. Automationshield. Online, 2019. [cit. 16.5.2019], <https://github.com/gergelytakacs/AutomationShield/wiki>.

- [13] Hernando Barragán. The Untold History of Arduino. Online, 2019. [cit. 18.5.2019], <https://arduinohistory.github.io/>.
- [14] J. Hurtuk, M. Chovanec, and N. Ádam. The Arduino platform connected to education process. In *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*, pages 000071–000076, Oct 2017.
- [15] H. Kovačević and Z. Stojanović. Buck converter controlled by Arduino Uno. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1638–1642, May 2016.
- [16] J. Li. Control System Laboratory with Arduino. In *2018 International Symposium on Computer, Consumer and Control (IS3C)*, pages 181–184, Dec 2018.
- [17] MakeltFrom.com. AISI 301 (S30100) Stainless Steel. Online, 2019. [cit. 4.5.2019], <https://www.makeitfrom.com/material-properties/AISI-301-S30100-Stainless-Steel>.
- [18] MATERIALPRO 3D. ABS vs. PLA vs. PET-G. Online, 2019. [cit. 26.3.2019], <https://www.materialpro3d.cz/blog/rozdily-abs-pla-petg/>.
- [19] Metal suppliers online. Stainless Steel 301. Online, 2019. [cit. 4.5.2019], <https://www.suppliersonline.com/propertypages/301.asp>.
- [20] MicroStep. Rezanie laserom / Technológie. Online, 2019. [cit. 27.4.2019], <https://www.microstep.sk/technologie/rezanie-laserom/>.
- [21] Mouser. F98 Series. Online, 2019. [cit. 27.4.2019], https://www.mouser.sk/datasheet/2/40/AVX_F98\%20DTSH\%2008_16-1158749.pdf.
- [22] M. Novák, J. Kalová, and J. Pech. Use of the Arduino Platform in Teaching Programming. In *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)*, pages 1–4, Oct 2018.
- [23] O. K. Ogidan, K. O. Temikotan, and K. C. Chike. Development of an Arduino microcontroller-based automatic load shedding module for teaching and research. In *2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, pages 1130–1134, Nov 2017.
- [24] C. A. Petry, F. S. Pacheco, D. Lohmann, G. A. Correa, and P. Moura. Project teaching beyond Physics: Integrating Arduino to the laboratory. In *2016 Technologies Applied to Electronics Teaching (TAEE)*, pages 1–6, June 2016.
- [25] ScienceABC. How Does a Zener Diode Work? Online, 2019. [cit. 5.5.2019], <https://www.scienceabc.com/innovation/zener-diode-work.html>.
- [26] Sparkfun. Spectrasymbol. Online, 2019. [cit. 5.4.2019], <https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/FLEX\%20SENSOR\%20DATA\%20SHEET\%202014.pdf>.

- [27] Stratasys. What is FDM Technology? Online, 2019. [cit. 26.3.2019], <https://www.stratasys.com/fdm-technology>.
- [28] Stredná priemyselná škola, Košice. REZISTORY. Online, 2019. [cit. 23.3.2019], <https://www.spseke.sk/web/kabinety/ele/rezistory.pdf>.
- [29] G. Takács, T. Konkoly, and M. Gulán. Optoshield: A low-cost tool for control and mechatronics education. In *Proceedings of the 12th Asian Control Conference (ASCC2019), June 9-12, 2019, Kitakyusu, Japan.*, pages 1–6, Kitakyushu, Japan, Jun 2019. Preprint submitted for review.
- [30] Technopedia. Rectifier. Online, 2019. [cit. 10.4.2019], <https://www.techopedia.com/definition/681/rectifier>.
- [31] Texas Instruments. Industry-Standard Dual Operational Amplifiers. Online, 2019. [cit. 10.4.2019], <http://www.ti.com/lit/ds/symlink/lm358a.pdf>.
- [32] Vangie Beal. API - application program interface. Online, 2019. [cit. 19.5.2019], <https://www.webopedia.com/TERM/A/API.html>.
- [33] VISHAY. Surface Mount Glass Passivated Junction Fast Switching Rectifier. Online, 2019. [cit. 27.4.2019], <https://www.mouser.sk/datasheet/2/427/rgf1-280141.pdf>.
- [34] WhatIs. Capacitor (capacitance). Online., 2019. [cit. 10.4.2019], <https://whatis.techtarget.com/definition/capacitor-capacitance>.